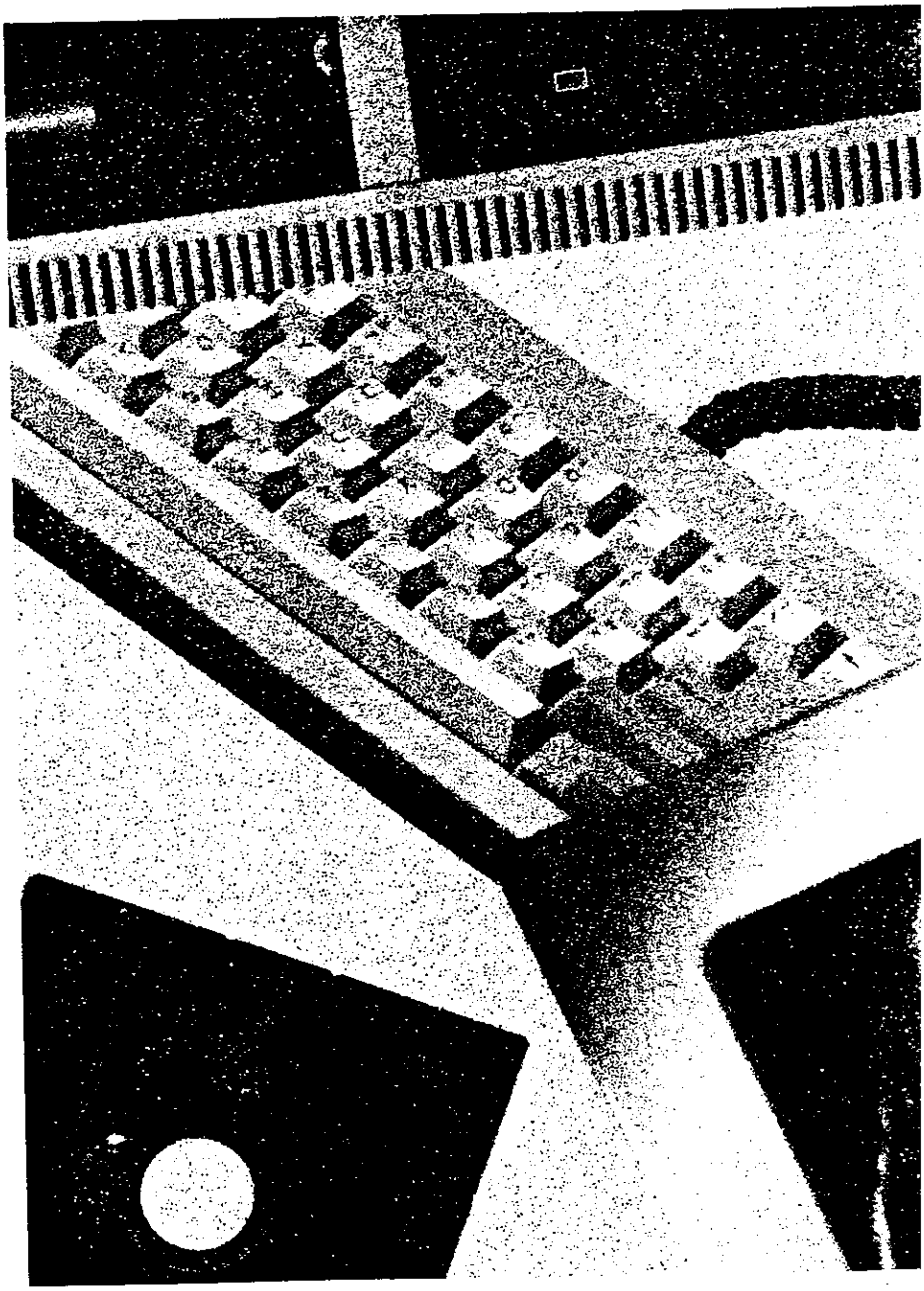
Enhanced MS-DOS 3.2 User's Guide



WYSE

Copyright Notice

©1987 Wyse Technology. ALL RIGHTS RESERVED.

This manual and the software and firmware described in it are copyrighted by Wyse Technology. You may not reproduce, transmit, transcribe, store in a retrieval system, or translate it into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, any part of this publication without the express written permission of Wyse Technology.

Based on a Tec–Ed document

Trademarks

WYSE is a registered trademark of Wyse Technology.

IBM, IBM PC, IBM PC/XT, IBM PC AT, and IBM PC DOS are registered trademarks of International Business Machines.

Microsoft, MS-DOS, and GW-BASIC are registered trademarks of Microsoft Corporation.

Ashton-Tate is a registered trademark of Ashton-Tate Corp.

AST is a registered trademark of AST Research, Inc.

Intel is a registered trademark of Intel Corp.

Lotus is a registered trademark of Lotus Development Corp.

Quadram is a registered trademark of Quadram Corp.

Disclaimer

Wyse Technology makes no representations or warranties regarding the contents of this document. We reserve the right to revise this document or make changes in the specifications of the product described within it at any time without notice and without obligation to notify any person of such revision or change.

Reference Guide

If you require more advanced information about Enhanced MS-DOS, we recommend that you refer to Wyse's *Enhanced MS-DOS Reference Guide*. This book is a complete reference covering all of the operating system's commands and the advanced utility programs provided with Enhanced MS-DOS. To obtain this book, contact your dealer.

Overview

The Enhanced Microsoft Disk Operating System (Enhanced MS-DOS) software is a collection of programs and commands that control the operation of your computer hardware. This operating system gives you access to the rapidly growing library of popular spreadsheet, database, and word-processing packages, as well as packages for accounting, communications, graphics, and much more.

The version of MS-DOS you received contains many commands and enhanced features that are not available in other forms of MS-DOS. Enhanced features include online HELP information, support for large-capacity hard disks (greater than 32 megabytes), system locking, reorganization of hard-disk files for improved performance with the DISKPACK command, caching of disk contents into memory for faster access, the ability to run EMS- and EEMS-compatible software in extended memory, and commands that control special hardware features.

The Purpose of This User's Guide

This user's guide is intended for persons who have just assembled and installed their computer hardware, have run the SETUP program according to instructions in the computer's installation and assembly manual, and are now ready to install and run application software. This guide assumes that the reader has little or no familiarity with MS-DOS.

The main purpose of this user's guide is to help you install MS-DOS and get your applications up and running as quickly as possible. A secondary purpose is to provide instructions for the MS-DOS operations you are most likely to perform frequently.

The guide introduces MS-DOS commands and procedures, and gives examples of techniques for using commands efficiently. However, because the guide is focused on getting you started, it doesn't try to show you everything you can do with MS-DOS. If you want to become an MS-DOS expert, you'll need to refer to sources of additional information, such as Enhanced MS-DOS's HELP screens or commercially available books.

Some of the commands and features of Enhanced MS-DOS are not present in other versions of the operating system. Experienced MS-DOS users can turn directly to Chapter 7 to learn how to apply these special capabilities.

How To Use This Guide

The information in this user's guide serves several purposes, so you can use it in different ways:

- To install MS-DOS and get your application up and running quickly, follow all the instructions in Chapter One that apply to the type of system you have.
- For a general overview of MS-DOS, read Chapter 2 before continuing with the procedures in the rest of the guide.
- As you have time, use Chapters 3 through 7 and Appendixes B and C as tutorial chapters. The sample commands printed in examples are suitable for practicing at your computer.
- Appendixes A and D are for your reference. For further reference information, the HELP command, described in Chapter 2, provides online explanations of commands.

As You Read This User's Guide . . .

Here are some points to remember as you read the instructions in this user's guide:

1 The term *disk* often refers generically to both diskettes and hard disks. Whenever the meaning is specific to one or the other, the appropriate term is used.

2 The names of keys appear as simulated keycaps in the text; for example, [Enter] and [Esc].

3 When you enter a command line or a response, you can type any combination of uppercase or lowercase characters and get the same results. In this guide, though, examples of commands and responses appear in UPPERCASE characters. Everything you type appears in **BOLDFACE** type. For example:

```
A>DATE
```

4 For the system to process any information you type, you need to press [Enter]. Throughout this guide, the word *enter* means *type the information or command and press* [Enter].

Table of Contents

Overview

1 Installing and Running Software

Getting Ready 1-1
Starting Up MS-DOS 1-3
Copying Your MS-DOS Diskettes 1-5
Preparing the Hard Disk 1-6
Formatting Diskettes 1-10
Installing Your Application Software 1-11
Starting and Leaving Application Software 1-13
Now That You're on Your Way 1-13

2 The Fundamentals of MS-DOS

Fundamental MS-DOS Concepts 2-1
Getting Online Help 2-2
Controlling Screen Displays and Printing Functions 2-4
Referring to Other Devices with MS-DOS 2-5

3 Working with Files

Understanding Files and File Names 3-1
Working with Disk Directories 3-4
Specifying File Names with Wild Cards 3-5
Displaying the Contents of a File 3-6
Renaming Files 3-7
Deleting Files 3-8

4 Working with Multiple Directories

Understanding the Directory Structure 4-1
Creating a Directory Structure 4-4
Specifying the Directory Path 4-6
Displaying the Directory Structure 4-7
Changing from One Directory to Another 4-8
Removing a Directory 4-10

5 Managing Files on Disk

Learning More about Copying Files 5-1
Backing Up and Restoring Your Hard Disk 5-5

6 Modifying the Operating Environment

Changing the System Prompt 6-1
Redirecting Output 6-3
Configuring for a Serial Printer 6-5
Modifying Your System Configuration 6-6

7 Enhanced MS-DOS Special Features

Taking Advantage of Large-Disk Support 7-1
Creating a Disk Cache 7-5
Running EMS-Compatible Software with Extended Memory 7-6
Combining CACHE, VMEM, and VDISK in Extended Memory7-8
Improving the Efficiency of Hard-Disk Storage 7-9
Controlling the Processor Speed 7-11
Controlling the Liquid Crystal Display (LCD) 7-12
Activating the System's Combination Lock 7-14

Appendixes

A Command Descriptions A-1
B Batch Files B-1
C International Configurations C-1
D Messages D-1

Index

1

Installing and Running Software

Now that you've assembled and installed the components of your computer, the next step is to load your operating system and application software. This chapter shows you how to install the operating system, prepare diskettes and the hard disk (if you have one) for use, and get your application programs up and running. By performing these instructions, you'll be ready to go in a few minutes.

Not everything in this chapter will pertain to the type of system you have; for example, you might not have a hard disk. You can skip over any instructions that don't apply to your system.

Getting Ready

Before you start, you need a little background information about disk drive configurations and drive designators so you can follow the procedures in this chapter. Also, you need to set out the diskettes that contain your operating system and application programs.

Identifying the Disk Drive Configuration

Look at the configuration of your disk drives. Your computer is most likely one of these two basic disk configurations:

- **A dual-diskette system.** You have two diskette drives. This configuration enables you to keep your MS-DOS diskette in one drive and perform most of your operations on the diskette in the other drive.
- **A hard-disk system.** You have at least one diskette drive and a hard disk. You use your hard disk most of the time.

Other configurations are possible; for example, your computer system may have only one diskette drive and no hard disk, have an extra disk drive, or be connected to a network. However, these other configurations are variations on the two just mentioned, so you can find instructions in this chapter that apply to your system.

Recognizing Drive Designators

The instructions that tell MS-DOS how to execute a command are stored on the disk in one of your system drives. To locate these instructions, MS-DOS needs to know which disk drive contains them. You need to know, too, because you may have to direct MS-DOS to the appropriate drive when you enter a command. The operating system assigns a *drive designator* to each disk drive attached to the computer. Normally, the drive designator is a letter from A to Z.

Your computer hardware has some combination of drives A, B, C, and D. Here's how to tell which disk drive is which:

Drive Designator	Disk Drive
A	The first or only diskette drive.
B	The second diskette drive in a configuration with two diskette drives. On a system with one diskette drive, *drive B* is the alternate designator of the single drive. MS-DOS uses the alternate drive designator to keep track of operations that involve more than one diskette (for example, copying information from one diskette to another).
C	The first or only hard-disk drive (if one is installed).
D	The second of two hard disks (if it is installed).

Additional disk drives have designators ranging from E through Z; but for now, you need to know only the drive designators A through D.

Using Diskette Drives

Sometimes diskettes that work well on one drive don't work at all on another. That's because diskette drives differ from each other in terms of the density or compactness of the information they can *read* (retrieve) or *write* (record).

☐ **Note** This discussion refers to 5.25-inch diskettes. (MS-DOS 3.2 also supports 3.5-inch disk drives.)

Your computer system comes with diskette drives that are either *double-density* (48-tracks-per-inch with 360K capacity) or *high-density* (96 tracks-per-inch with 1.2MB capacity) drives. Both types of drives read from or write to both sides of the diskette. (The computer drives are not single-density or single-sided, although they can read the contents of diskettes produced by such drives.)

A double-density drive can read, write, and format double-density diskettes. A high-density drive can read, write, and format both double-density and high-density diskettes.

Caution When you format a diskette in a high-density drive, find out the density of the diskette media. If it is a double-density diskette, you must format it to have only a 360K capacity; otherwise errors will result.

After any diskette has been formatted on a high-density drive, or has been written on by a high-density drive, you may not be able to correctly read that diskette in a double-density drive.

Setting Out Your Diskettes

Your MS-DOS operating system comes on two diskettes. One of them contains the central portions of the operating system itself and a set of *user files* or commands; this guide refers to that diskette as the *MS-DOS diskette*. The other disk contains supplemental programs. Unpack these diskettes and set them next to your computer.

Certain instructions in this chapter call for you to copy information onto blank diskettes; for example, you will copy the MS-DOS diskette, and you may need to copy an application program from one diskette to another. You should have as many blank diskettes available as you have MS-DOS and application program diskettes combined.

Now you're ready to begin.

Starting Up MS-DOS

If your computer is off, you need to start (or *boot*) MS-DOS before you can do anything else. Insert the MS-DOS diskette into drive A, turn on the power, and close the drive lever.

If your computer is on but idle, make sure the MS-DOS diskette is in drive A and the drive lever is closed. Then press Ctrl Alt Del simultaneously (a *reboot* or *system reset*).

The computer performs an internal memory check and then the boot procedure begins.

Entering the Date

After a few seconds, the screen displays this message:

```
Current date is Wed 1-01-80
Enter new date (mm-dd-yy):
```

This message is a *prompt*; it tells you what kind of information to enter and how to enter it. If you have just finished running the SETUP program as described in your installation and assembly manual, the date shown should be correct; in that case, just press Enter. If the date is incorrect, respond to the prompt by typing today's date using the format shown within parentheses (you can omit leading zeros). For example:

```
Enter new date (mm-dd-yy): 4-29-87
```

If you make a typing mistake, press ←Backspace to return to the mistake and then correct the entry, retyping any other characters erased. Now press Enter.

Entering the Time

A similar message prompts you for the time of day:

```
Current time is 10:22:58.79
Enter new time:
```

If the time shown is correct, press Enter. If the time is incorrect, type the current time, separating the hours and minutes with colons (you don't need to specify seconds or hundredths of seconds). MS-DOS operates on a 24-hour clock; if you start up your system at 1:00 p.m., enter **13:00**.

☐ **Note** When you change the date and time as shown here, the new date and time remain in effect only until you turn off the computer or reboot MS-DOS. If you want the computer to boot up with the changed date and time, you must run the SETUP program as described in your installation and assembly manual.

Changing the Active Disk Drive

After you enter the date and time, MS-DOS displays copyright information, followed by this prompt:

```
A>
```

This symbol is called the MS-DOS *system prompt*; it means MS-DOS is ready to process commands from drive A (the *active* or *current* disk drive). Whenever you enter a command, MS-DOS looks on the drive shown in the prompt for the instructions needed to execute the command.

To change the active drive from drive A to another disk drive, enter the designator of the drive you want to change to, followed by a colon. For example:

```
A>B:
```

(Always be sure to press Enter after the last character you type.)

In this case, MS-DOS immediately displays:

```
B>
```

This system prompt indicates that drive B is now the active disk drive.

Before proceeding, make drive A the active drive again.

Copying Your MS-DOS Diskettes

Your first task as you prepare your system for use is to make backup copies of your MS-DOS diskettes. You'll use the copies for all further MS-DOS processing.

Your original copy of the MS-DOS diskette is already in drive A. For this procedure, it is the *source diskette*, because it contains the information you want to copy.

Insert a blank diskette (the *target diskette* onto which MS-DOS will copy the information) into drive B and close the lever. Then type the following command and press Enter:

```
A>DISKCOPY A: B:
```

This command tells the operating system to make an exact duplicate of the drive A diskette on the drive B diskette. When you press Enter, MS-DOS first prepares the blank diskette to receive information, and then displays a message showing the number of tracks, sectors, and sides of the diskette being copied.

(If your system has only one diskette drive, MS-DOS will prompt you during the DISKCOPY process to remove the source diskette and replace it with the target diskette, and to press any key when you're ready. Instructions of this type will appear on the screen at various times while DISKCOPY runs.)

After a moment, the operating system asks:

```
Copy another diskette (Y/N)?
```

Before you respond, remove the diskettes from both drives and attach a label to the diskette jacket that describes what the copied diskette contains; to avoid damaging the diskette's surface be sure to use only a felt-tipped pen. Next, insert the MS-DOS supplemental programs diskette into drive A and another blank diskette into drive B. Respond *yes* to the prompt.

Repeat this DISKCOPY procedure until you've made copies of every program diskette that was supplied with your computer. At that point, when DISKCOPY asks if you want to copy another diskette, answer *no*. The operating system will immediately return to the A> system prompt.

Now finish labeling the copied diskettes. For the rest of the procedures in this guide, use these copies. Put the original MS-DOS diskettes in a safe place.

☐ **Note** If you are using this computer outside the United States or are using a non-United States keyboard, refer to Appendix C to find how to produce an MS-DOS system diskette that will set up your computer for other national standards.

If you don't need to prepare a hard disk for operation, you can skip the next section and go on to "Formatting Diskettes" later in this chapter.

Preparing the Hard Disk

Preparing the hard disk for operation entails six procedures: initializing the hard disk, sparing out bad tracks identified on the hard disk's label, testing it for any additional defective tracks, partitioning it, formatting it, and copying MS-DOS files from the system diskette to the hard disk. At the end of these procedures, you will be able to boot MS-DOS from the hard disk rather than from the system diskette.

The disk-preparation procedures in your installation and assembly manual showed you how to initialize the hard disk and test it for bad tracks. This guide assumes that you've already completed those procedures. The following instructions show you how to partition and format the hard disk (with the FDISK and FORMAT commands) and how to copy MS-DOS onto it.

Caution If your hard disk has already been prepared and formatted, you may skip this section. The following procedure will destroy all files already on the disk.

Partitioning the Hard Disk

Before using the hard disk, you must allocate a section for use by MS-DOS. The FDISK command creates the partition you will require. The following procedure assumes you plan to use the entire hard disk as one large partition for MS-DOS.

Some versions of MS-DOS restrict the maximum size of a hard-disk partition to 32MB (megabytes). This enhanced version lets you create partitions larger than 32MB, up to the capacity of your disk.

☐ **Note** Advanced users can find more detailed information about large-disk support and multiple partitions in Chapter 7, "Enhanced MS-DOS Special Features." (If you plan to have another operating system make use of all or part of the hard disk, consult the reference manual for the other operating system you intend to use.)

To create a single MS-DOS partition on your hard disk, follow these steps:

1. Enter the FDISK command:

   ```
   A>FDISK
   ```

   The operating system displays a list of several choices affecting the partitions of a hard disk.

2. Select the option that creates an MS-DOS partition.

3. The operating system asks if you want to use the entire disk for MS-DOS (that is, to create one large partition rather than divide the disk into several smaller ones). Answer *yes*.

4. FDISK then displays information about the partition you have created. Return to FDISK main menu. (You don't have to deal with the other options on this menu if your hard disk will be a single partition.)

5. Return to the MS-DOS system prompt and reboot the operating system with your MS-DOS diskette in drive A.

6 When the operating system boots up, it partitions the hard disk as you instructed. Respond to the date and time prompts and the A> system prompt will reappear.

Now continue with the hard disk preparation procedure by formatting the hard disk.

☐ **Note** If you are using this computer outside the United States or are using a non-United States keyboard, refer to Appendix C to find how to format and prepare your hard disk so MS-DOS will set up your computer for other national standards. Follow those instructions instead of those in the next section on formatting the hard disk.

Formatting the Hard Disk

The format procedure for the hard disk is similar to the one you used for formatting diskettes. In this procedure, you simultaneously format the hard disk and copy the MS-DOS system files onto it.

1 Before you run the FORMAT command for the hard disk, you must know whether the disk already has a *volume label*. To find out, enter this command:

```
A>VOL C:
```

If a volume label (which may be up to 11 characters long) appears, write it down so you can type it when the FORMAT program asks you for that information.

2 Enter the FORMAT command like this:

```
A>FORMAT C: /S /V
```

The characters /S tell MS-DOS to copy its system files (COMMAND.COM and two hidden files) onto the hard disk. The /V indicates you want to give the disk a volume label.

3 If the hard disk already has a volume label, you'll now be asked to enter that name. Do so. If you enter the name incorrectly, FORMAT will return you to the system prompt; in that case, enter again the command given in step 2.

4 The operating system displays a warning message, stating that the format operation will erase any information already on the disk, and asking if you want to go ahead with the formatting. Since the hard disk contains no data yet, indicate that you *do* want to continue.

5 When the formatting process is finished, you'll be asked to enter an 11-character volume label for the hard disk. This label may contain letters, numbers, and spaces between words, but no tabs or any of the following symbols:

. " / \ [ ] : | < > + = ; ,

(The label need not be a name that the disk had previously.)

6 When the formatting process finishes, MS-DOS displays a message telling the amount of memory available on the hard disk, and then returns to the A> prompt.

Although this procedure lets you boot MS-DOS *system* files from the hard disk, it doesn't copy all of the MS-DOS *user* files that you need. To copy these user files, go on to the next procedure.

Copying the MS-DOS User Files onto the Hard Disk

To copy the contents of the first MS-DOS system diskette from drive A to drive C, enter the COPY command like this:

```
A>COPY A:*.* C:
```

(The *.* symbol means *all files*.) The operating system copies the files on the MS-DOS diskette to the hard disk, and then returns to the A> prompt.

Now remove the diskette from drive A and replace it with the MS-DOS supplemental programs diskette. Next, reenter the COPY command exactly as before. Once again, the operating system copies the files and returns to the A> prompt. Finally, remove the diskette from drive A.

This procedure completes the setup of the hard disk. You can now boot MS-DOS from the hard disk, instead of using the MS-DOS system diskette.

Booting MS-DOS from the Hard Disk

To boot from the hard disk, be sure no diskette is in drive A. Press Ctrl Alt Del simultaneously. The operating system checks whether drive A contains a system diskette; then it looks for the system files on drive C and boots up from there. After a moment, it prompts you for the date.

If necessary, correct the date and time; otherwise, press Enter twice. When you finish, the operating system displays a new system prompt:

```
C>
```

When you successfully complete this process, your hard disk contains all the MS-DOS files. You no longer need MS-DOS diskettes to perform operations.

Formatting Diskettes

Unless you use DISKCOPY to prepare a new diskette, you must always format new diskettes with the FORMAT command before using them for the first time.

You can also use the formatting procedure to create diskettes that are able to boot the operating system (such diskettes don't require all the user files provided on the original MS-DOS diskette). If you store your application programs on such "bootable" diskettes, you don't need to insert the MS-DOS diskette first to start the operating system.

☐ **Note** The following procedure formats diskettes for use on a drive of the same capacity as the formatting drive; that is, formatting on a high-capacity drive prepares 1.2MB diskettes, while formatting on a double-density drive prepares 360K diskettes. If you want to use a high-capacity drive to format a double-density diskette, pay attention to the note in step 2.

Follow these procedures to format blank diskettes into bootable diskettes:

1 If your system has no hard disk, insert your copied MS-DOS diskette in drive A.

2 Enter the FORMAT command like this:

```
A>FORMAT B: /S /V
```

(If your system has a hard disk, the prompt here should be C> rather than A>; the hard disk, drive C, is the active drive.)

☐ **Note** If you want to use a high-capacity drive to format a double-density diskette (360K), you should include the /4 parameter in the FORMAT command:

```
FORMAT B: /4 /S /V
```

The /4 must precede /S and /V in the command.

The characters /S tell MS-DOS to copy its system files (i.e., COMMAND.COM and the operating system's two hidden files) onto the diskette in drive B. The characters /V tell

MS-DOS that you want to record a *volume label* or diskette name on the diskette. (You don't always have to include the /S and /V parameters in the FORMAT command, because not every diskette needs the system files or a volume label.)

3 If you have a dual-diskette system, insert a blank diskette in drive B. If you have a single-drive or hard-disk system, replace the MS-DOS diskette with a blank diskette (the operating system thinks the drive is now called *drive B*). When you're ready, press Enter.

The operating system displays messages that indicate the process of the formatting, which takes about a minute per diskette.

4 When this process ends, the operating system prompts you to enter a label or diskette name of up to 11 characters (this step takes place because you included the /V parameter in the FORMAT command).

Enter the text of the label. You can use letters, numbers, and spaces between words; however, you cannot use tabs or the following symbols:

. " / \ [ ] : | < > + = ; ,

For a diskette containing letters written during March of 1987, you could enter this volume label:

```
LTRS MAR 87
```

5 Remove the formatted diskette from the drive and set it in a place away from any remaining unformatted diskettes.

6 You are now offered the choice of formatting more diskettes or of ending the process and returning to the MS-DOS system prompt.

Installing Your Application Software

Some business applications come with a diskette containing installation software that automatically prepares the formatted disk for the application and copies the files. To install the application, you should follow its packaged instructions

For applications without installation software, check the packaged instructions for information on any special installation procedures, such as creating directories. This section assumes that no special procedures are necessary.

The procedures in this section explain how to copy application programs from a source diskette to a target diskette or to the hard disk. Copying your application software onto a second set of diskettes provides you with a backup set, which you should then use to run the application. Copying the software onto the hard disk gives you convenient access to the software.

Caution Before you copy anything, though, find out whether your application diskettes are copy-protected. If they are, read the packaged instructions carefully to determine what this protection requires you to do.

When you're ready to install the application software, put a write-protect tab on the original diskette and insert it in drive A. (This will be considered the source diskette.) Then follow the instructions below that apply to your type of system. When you finish, be sure to label all copied diskettes, and store the originals in a safe place.

- **For a dual-diskette system.** Place a blank, formatted diskette (the target diskette) into drive B. Then enter the COPY command like this:

  ```
  A>COPY A:*.* B:
  ```

  (The *.* symbol means *all files*.) The operating system copies all files from drive A to drive B, displaying the name of each file copied. When the procedure ends, MS-DOS returns to the A> prompt.

  Remove the source diskette from drive A and replace it with the next source diskette, if there is one. Do the same with the target diskette in drive B. Then reenter the COPY command exactly as before.

  Repeat the procedure until all application diskettes are copied.

  Note If you have a single-drive system, you can enter the COPY command as shown here. MS-DOS prompts you to exchange source and target diskettes until the copy operation ends.

- **For a hard-disk system.** With the source diskette in drive A, enter the COPY command like this:

  ```
  C>COPY A:*.* C:
  ```

The operating system copies all files from drive A to drive C, displaying the name of each file copied. When the procedure ends, MS-DOS returns to the C> prompt.

Remove the source diskette from drive A and replace it with the next source diskette, if there is one. Then reenter the COPY command exactly as before.

Repeat the procedure until all application diskettes are copied.

Starting and Leaving Application Software

When you run application software, refer to its manual to find how to start up the program.

You should leave the application by first completing all operations within the program; then enter the program's command for exiting back to the operating system. At that point the MS-DOS prompt reappears, which allows you to start other programs or perform MS-DOS functions.

Now That You're On Your Way . . .

MS-DOS has many functions that enhance the convenience of running applications on your computer system. This user's guide contains practical tips about using MS-DOS, managing and manipulating files, and modifying the ways your computer system handles input and output (the *operating environment*).

Of course, MS-DOS has much more to offer than this user's guide can describe fully. Appendix A provides summary information about MS-DOS commands. If you want more information about operating system commands, you can refer to the online HELP command or consult one of the many MS-DOS books and reference manuals commercially available.

2

The Fundamentals of MS-DOS

In Chapter 1, you entered MS-DOS commands as directed step-by-step to install the operating system and perform other tasks. Now that you've installed MS-DOS, you'll learn to use its commands in various contexts that demand greater flexibility. This chapter provides background information about MS-DOS that applies to all of the operating system's commands.

Fundamental MS-DOS Concepts

MS-DOS is an acronym for *Microsoft Disk Operating System*. Designed for your personal computer by Microsoft Corporation, the operating system's primary role is to help you run your application programs by controlling the operation of the computer system.

MS-DOS begins this activity as soon as you start any computer program. After the program starts, MS-DOS fades into the background. It's as active as ever, executing commands that the program sends it; but you seldom see that activity, and may not be aware of MS-DOS until the program ends.

The central part of MS-DOS is stored on disk in three operating system files: COMMAND.COM and two *hidden files* that aren't identified by name in a directory listing. COMMAND.COM is called a *command processor* file. It tells the computer how to process any command you enter and contains an internal set of commands. Whenever you turn on or reset the computer, these three system files automatically load into main memory and start running.

Types of Commands

The command processor of MS-DOS recognizes two types of commands: those that exist as instructions within the operating system (*internal commands*), and others that are stored in program files distinct from the operating system files (*external commands*).

To run an external command, the program file for that command must be on the currently active disk drive, or on another drive you specify along with the command. When you enter an external command, MS-DOS looks on the active or specified drive for a program file with the command name, and then executes the file. (FORMAT is an example of an external command.)

Messages

MS-DOS displays two types of messages on the monitor screen: system messages and error messages.

System messages provide information about the state of the computer system or about an operation just completed. *Error messages* tell you that the system can't continue processing for some reason.

Sometimes, system and error messages merely provide information. At other times, they prompt you to do something. Appendix D, "Messages," lists and explains the messages.

Getting Online Help

Enhanced MS-DOS has an online HELP feature that provides some reference information about each command.

HELP is an external command. To use the HELP feature, be sure a file called HELP.EXE is on the disk in the active drive.

To find out which commands are available with your version of MS-DOS, enter the HELP command:

```
A>HELP
```

MS-DOS responds by displaying a list of all the available commands, and then returns to the A> prompt.

The HELP feature serves as an on-the-spot reference for commands. To use this reference, enter the HELP command followed by the name of the MS-DOS command you want information about. For example, when you enter this command

```
A>HELP ASSIGN
```

MS-DOS displays a summary of the ASSIGN command's features.

As you work through the procedures in this guide, you should occasionally experiment with HELP by looking up the command you're using.

Accessing Online Help from a Hard Disk

If you have a hard-disk system, the HELP command's on-line explanations of MS-DOS commands can be made available while you are working in any drive and directory. Convenient access to HELP is established with the PATH command.

The PATH command allows you to specify directories that will be searched automatically for any command or program file that is not in the current drive's current directory (which is always the first to be searched). For example, this command:

```
PATH C:\
```

tells the operating system to search in the root directory of drive C. In another example, this command:

```
PATH C:\;C:\DOS
```

tells the operating system to search first in the root directory of drive C (C:\) and then in a subdirectory named DOS on drive C (C:\DOS). Semicolons are used to separate the multiple directories listed in a PATH command.

Creating the Path to Help

You can enter a PATH command that will make on-line help available at all times. For example, if the file needed to execute the HELP command (HELP.EXE) is located in the root directory of drive C, enter this command:

```
PATH C:\
```

(If HELP.EXE is in another directory, include that directory's name in the PATH command.) From then on, while working in any current drive and directory, you can request information on any MS-DOS command with a HELP command; for example:

```
HELP CHKDSK
```

A PATH command such as the above can be executed automatically every time you boot up the computer if that PATH command line is contained in an AUTOEXEC.BAT file. A method for creating an AUTOEXEC.BAT file is explained in Appendix B.

Controlling Screen Displays and Printing Functions

Your operating system provides several useful functions for controlling operations such as screen displays and printing. These functions let you cancel an operation in progress, control scrolling (the vertical movement of text on the screen), print the contents of a screen display, and print a record of the on-screen interactions between yourself and the computer system.

The following table lists and describes these functions, identifies the keys you press to start or stop them, and describes the results.

Function	Description	Keys You Press	What Happens
Break	Cancels an operation in progress.	Ctrl Break	Stops the operation and returns to the system prompt.
Scroll control	Temporarily interrupts the scrolling of text.	Pause or Ctrl Num Lock	"Freezes" the screen image. Press any key to resume scrolling.
Print screen	Sends to the printer each line now on the screen.	Print Screen or ⇧ Shift PrtSc	Prints a duplicate of the screen image.
Printer echo on/off	Switches the printer between printing and not printing input and output that subsequently appears on the screen.	Ctrl Print Screen or Ctrl PrtSc	Any text that appears on the screen after you press these keys will be printed by the printer. Press these keys again to stop the action.

Referring to Other Devices with MS-DOS

Your operating system manages several devices connected to the computer by cables attached to special sockets called ports. To MS-DOS, a *device name* is actually the name of the port to which the device is connected.

The standard device names are:

Name	**Description**
AUX	Abbreviation for *auxiliary device*; refers to the first serial port (COM1) by default.
COM1 and COM2	The first and second serial *communications ports*, respectively (COM1 is the same as AUX). These ports accommodate serial input devices such as a mouse, touch pad, and optical character reader; serial output devices such as printers and plotters; and serial input/output devices such as modems and connections to other computers.
CON	Abbreviation for *console*; refers jointly to the keyboard and the monitor.
LCDDEV	The 16-character LCD panel, available on some systems. Before MS-DOS will recognize this device name, however, the LCDPANEL.SYS device driver must be loaded by your including it in the CONFIG.SYS file (see "CONFIG.SYS Commands" in Chapter 6 for details).
LPT1, LPT2, and LPT3	Abbreviation for *line printer ports*, for the parallel printers attached to the first, second, and third parallel ports.
NUL	A "dummy" device, used for testing applications.
PRN	Abbreviation for *printer port*, the first parallel port (PRN is the same as LPT1).

Several commands in this guide call for these standard device names. For example, the MODE command, described in Chapter 6, refers to printers and communication devices. With other commands, such as COPY, you can include device names that identify the source or destination for input or output.

3

Working with Files

Because most of your work with MS-DOS will involve files, this guide devotes three chapters to file operations.

Most operations involve file management, such as handling file directories, displaying file contents, and renaming and deleting files. This chapter provides some background information about files, and then describes these basic file management operations. The two chapters that follow deal with more advanced file management, which involves handling directories and subdirectories.

If you want experience entering MS-DOS commands at your computer, you can create a practice diskette from any disk containing files that you might have. (Follow the DISKCOPY or COPY *.* procedures explained in Chapter 1 to make a practice copy of your source diskette.)

In this chapter and those that follow, we provide many examples of command entries in which we mention fictitious file names. When you enter the sample commands you see in this user's guide, substitute appropriate real file names from your practice diskette in place of our fictitious file names.

Understanding Files and File Names

Disk files generally fall into two categories: program files and data files.

Program files contain instruction codes readable only by the computer. You don't access the contents of program files directly—you use them as tools to work on data files. *Data files* provide the information (text and numbers or graphic data) that the program files need to do their work. Data files require an operating system command or an application program to access the file contents.

Types of Program and Data Files

These two file categories break down into special file types. The way you use a file depends on its file type:

Program File Type	Description	Sample Uses
MS DOS command file	Contains instructions that MS-DOS executes when you enter the command file's name	Management of all files on disk, file input and output
Application program	Instructions for performing specialized tasks that apply to your business or personal goals	Word processing, communications, other business functions
Batch file	A list of MS-DOS commands that the operating system executes in sequence when you enter the file name	Simplifying entry of routine or complex command sequences

Data File Type	Description	Sample Use
Text (ASCII) file	Information consisting of ASCII codes that define number and letter characters	Text display, word processing
Binary (non-ASCII) file	Information stored in a nontext format	Graphics, sound output, binary data

Assigning File Names

The name of an MS-DOS file has two components: the name proper and its extension.

The file name proper consists of one to eight characters with no spaces between them. It usually describes the contents of the file in some way; for example, LETTERS, LTS386, or LTS(MAR). The file name may contain any keyboard characters *except*:

. " / \ [ ] : | < > + = ; ,

These characters have special meanings to MS-DOS. If you attempt to include these characters in a filename, you'll see an error message. You also cannot use device names (such as CON and PRN) as file names.

The extension is a suffix to the file name, and describes the file type in some way. A period separates the extension from the file name, as in LETTERS.TXT. The extension consists of up to three characters, but otherwise follows the same rules as file names.

For most files you create, you are not required to include extensions in their names, but they are advisable. They provide a convenient way to recognize related files and deal with them as a group. MS-DOS lets you use the same name for two files *if* their extensions are different.

Although you can use nearly any three-character combination in a file extension (as long as you avoid the characters not permitted in file names), the following extensions appear often on disks containing software:

Extension	Meaning	Importance for You
.BAS	A program written in BASIC	You must load the BASIC language to run this program.
.BAT	A batch file, which is a sequence of MS-DOS and other commands	You run the file by entering the file name (you do not need to enter the .BAT extension).
.COM	A command (or program) file	You can execute the command (or program) within MS-DOS by entering the file name (you do not need to enter the .COM extension).
.EXE	An executable program file	You can execute the program (or command) within MS-DOS by entering the file name (you do not need to enter the .EXE extension).
.SYS	A system file or a device driver	The principal .SYS file is CONFIG.SYS, which sets up the computer's configuration each time you turn it on. CONFIG.SYS may contain commands that load other device driver files (e.g., DEVICE = VDISK.SYS).

Aside from these file name extensions that have specific meanings within MS-DOS, many application programs assign special extensions to the files they create or access. For example, .OVR may identify overlay files, .WKS may identify worksheets, and .BAK may indicate a backup file.

Working with Disk Directories

A *directory* is a list of files on a disk. Programs use this list to locate files. With the DIR command you display a listing of this directory to obtain information about the files in it such as their file names and extensions, sizes (in bytes), and the dates and times files were created or last updated.

Displaying All the Files in the Directory

To list information about all the files in a directory, use the DIR command. For diskette drives, insert a diskette into the current drive and enter the command. (If the hard disk is the current drive, simply enter the command.) For example:

```
A>DIR
```

To display the directory listing of a disk in a disk drive other than the current drive, enter the DIR command followed by a space and the drive designator; for example:

```
A>DIR C:
```

The system scrolls a list of all the files (except "hidden" files) that are on the drive designated in the command. The following example shows a typical directory listing:

```
Volume in drive A is WORDPROCESS
Directory of A:

COMMAND    COM    23612     3-21-86    12:00p
ANSI       SYS     1651     3-21-86    12:00p
FIND       EXE     6403     3-21-86    12:00p
WP         COM   171509     1-17-87     1:43p
WPHELP     OVR    87340     1-17-87     1:12p
FORM1      LTR    10091     1-13-87     8:59a
FORM2      LTR     5577     1-07-87     3:27p
WEST       ADR     3727     3-07-87     2:41p
FORM2      BAK     5231    12-16-87     4:41p
FORM1      BAK    12307    11-12-87    10:34a
           10 File(s)        134948 bytes free
```

If the disk has a volume label (e.g., WORDPROCESS), the directory listing displays it. If it didn't have a name, the message would say that the disk has no name (i.e., "Volume in drive A has no label").

The file names and extensions are in the left two columns; note that the system doesn't display the period separator before the extension. The third column is the file size in bytes. The two columns on the right show the date and time of file creation or update.

The last line of the listing indicates the number of files on the disk and the amount of disk space remaining.

Here are some other ways to use the DIR command:

To List	You Enter	Result
The complete directory listing one screenful at a time	DIR /P	Lists directory entries until the screen is full and then pauses (Press any key to display the next screenful of entries.)
File names and extensions only	DIR /W	Creates a "wide listing" of names and extensions (but no other file information) in five columns across the screen
Information about one file on the current drive.	DIR FORMAT.EXE	Displays the information for the file named in the command
Information about one file on a noncurrent drive	DIR B:FORMAT.EXE	Displays the information for the file named in the command

Specifying File Names with Wild Cards

Suppose you want to know the names of all the files on the disk that have the "EXE" file name extension. With DIR and many other commands, MS-DOS provides a feature that enables you to indicate those files as a group.

A *wild card* is a special character that substitutes for one or more unknown characters. MS-DOS has two wild cards: the question mark (?) and the asterisk (*).

Substituting for One Unknown Character

The ? wild card substitutes for any one unknown character in a name. A name with no character in that position will also satisfy the specification. The ? wild card can appear anywhere in the file name or extension. For example, if you enter this command:

```
A>DIR BASIC?.BAT
```

the operating system lists files called BASIC.BAT and BASICA.BAT, but not BASICAB.BAT. If you want the listing to include all three files, you need to use two questions marks (i.e., BASIC??.BAT), which would allow matches with two unknown characters.

Substituting for Multiple Unknown Characters

The * wild card substitutes for all unknown characters from the current position to the end of the file name or the end of the extension. For example, if you ask MS-DOS to locate files with the name BASIC.*, the system displays all files with the file name BASIC, regardless of extension.

Similarly, if you enter the DIR command this way:

```
A>DIR *.EXE
```

MS-DOS displays a directory listing of all the files on the disk that have an .EXE file name extension. Entering the command DIR *.* displays all file names on the disk, as if you had entered DIR alone.

☐ **Note** If you type the * wild card at the beginning of a file name or extension, followed by additional characters, MS-DOS ignores the other characters and assumes that you want to list all file names or extensions. For example, typing **DIR *SIC.COM** results in a display of all files with a .COM extension, not just those with *SIC* as the last three letters in the file name.

Combining Wild Cards in File Names

You may include several wild cards in the same file name. For example,the command **DIR BASIC?.*** tells MS-DOS to locate all files with BASIC as the first five characters, regardless of the extension or the sixth character in the file name.

You can use wild cards in commands other than the DIR command. As you saw in Chapter 1, the command **COPY *.*** means "copy all files on the disk."

Displaying the Contents of a File

If you want to see what an ASCII text file contains without changing the file, use the TYPE command followed by the file name and extension. For example, to look at the text of a letter to John Smith in a file called SMITH.LTR, enter the command:

```
A>TYPE SMITH.LTR
```

MS-DOS looks on the current drive for the file and displays its contents on the screen. If the file you want is on another drive, include the drive designator before the file name.

Always specify the file extension, if any, with TYPE. If the disk contains two files with the same name but different extensions and you don't include a specific extension in a command to TYPE one of the files, the operating system will display a message saying that it can't find the file.

When MS-DOS finds the file, it displays the entire contents of the file on the screen. If the file is longer than 24 lines, the operating system scrolls the text up the screen and the earlier lines vanish from sight.

You can tell MS-DOS to display long files one screenful at a time by attaching the MORE command to the end of the TYPE command entry. Begin the TYPE command as usual; then add a vertical bar character (¦) and the word MORE.

For example, to display SMITH.LTR one screenful at a time, enter:

```
A>TYPE SMITH.LTR ¦ MORE
```

MS-DOS displays the first screenful of text, followed by

```
-- MORE --
```

at the bottom of the screen. Press any key to display the next screenful of text.

TYPE doesn't accept wild-card characters. You must enter the complete file name.

Although MS-DOS does try to display the contents of a program file or nonASCII data file if you ask it to, what you see may look like a jumble of symbols. Don't worry: The TYPE command causes no harm to the file.

Renaming Files

The RENAME (or REN) command enables you to change the name of one file or a group of files. To rename a specific file, type the command followed by the drive designator (if necessary) and the current name and extension, then the new name and extension. (The new name doesn't need a drive designator.) For example, to change a file's name from SMITH.LTR to SMITH.DOC, enter this command:

```
A>REN SMITH.LTR SMITH.DOC
```

If the new name already belongs to another file, the operating system displays an error message to that effect and returns to the system prompt; in that case, reenter the command using a different file name.

REN accepts wild-card characters in both the current name and the new name. You can use wild cards as a shorthand way to change the file name and the extension of one file or of groups of files. Here are some possibilities:

You Want To	You Enter This Command	Original Name	Resulting Name
Change only the specified part of the file name, leaving remaining characters unchanged	REN SMITH.DOC SMY*.*	SMITH.DOC	SMYTH.DOC
Rename several files, keeping the same extension and wildcard character	REN SMITH?.DOC CLIENT?.DOC	SMITH1.DOC SMITH2.DOC SMITH3.DOC	CLIENT1.DOC CLIENT2.DOC CLIENT3.DOC
Rename several files, changing both the name and the extension, keeping same wildcard character	REN SMITH?.DOC CLIENT?.LTR	SMITH1.DOC SMITH2.DOC SMITH3.DOC	CLIENT1.LTR CLIENT2.LTR CLIENT3.LTR
Change all files with one file extension to another file extension, keeping the same file name	REN *.DOC *.LST	MAIL.DOC NAMES.DOC CLIENT.DOC PROSPECT.DOC	MAIL.LST NAMES.LST CLIENT.LST PROSPECT.LST

Be careful about renaming groups of files with wild cards. You may accidentally rename files whose names should be left unchanged. Before renaming groups of files, always use the DIR command to list the names of files that the REN command might affect.

Deleting Files

You can delete files to remove unwanted information, to clear space on the disk, or for a variety of other reasons. Deleting a file means removing it from the file directory. Although the file contents remain on the disk (until the disk space they occupy is taken up by a new file), the drive can't locate them and considers the space they occupy to be vacant.

The DEL and ERASE commands are synonymous. Using either command, you can delete files singly or in groups.

To delete a single file, enter the DEL or ERASE command followed by the drive designator (if necessary) and the file name and extension; for example:

```
A>DEL TEST1.PLA
```

The operating system immediately deletes the file from the diskette in drive A and returns to the system prompt.

You can use wild cards with the DEL and ERASE commands. For example, to delete all files with a .PLA extension from the diskette in the active drive, enter the command:

```
A>DEL *.PLA
```

As before, the operating system immediately deletes the files.

For individual and group file deletions, MS-DOS doesn't give you the chance to change your mind after you enter the DEL (or ERASE) command. Be extremely careful when you use wild cards to delete files—you can easily delete files you want to keep. Once you delete a file, you can't "undelete" it. Before deleting groups of files, always use the DIR command to list the names of files that the DEL command might affect.

To delete all files from the disk, enter the command:

```
A>DEL *.*
```

Only for such a global deletion does MS-DOS stop to ask you to confirm that you really want to delete all files.

4

Working with Multiple Directories

In Chapter 3, "Managing Files with MS-DOS," you saw how to display the disk's directory listing with the DIR command. The discussion assumed you had only one directory on the disk—the main or *root* directory. With MS-DOS, you can create and maintain other disk directories that branch off from the root directory.

As the second of three chapters on file management, this chapter begins by explaining the concept of a directory structure. It then shows how to create a disk with several levels of directories, follow the *path* to a specific directory, display the current directory structure, change activities from one directory to another, and remove a directory from the disk.

Understanding the Directory Structure

A few of the commands you've already used have displayed messages that refer to the root directory of a disk. This directory is the one you've been using up to now.

MS-DOS also lets you make *subdirectories*, each of which can have its own subdirectories.

Recognizing the Root Directory and Subdirectories

The root directory is the main directory of a disk. (On most diskettes and some hard disks, it's the only directory.) It contains entries about files consisting of file names, extensions, sizes, locations on the disk, and so on.

To MS-DOS, a subdirectory looks like any other file in the root directory. The root directory lists it just like any file, except that the subdirectory's listing has the label <DIR> after its name. In fact, a subdirectory *is* a file—a file that contains directory entries for files not in the root directory.

To find a file in a directory, MS-DOS must access that directory; otherwise, it doesn't know the file exists. In effect, a directory and its subdirectory represent completely separate storage areas on the disk, because MS-DOS acts much as if the files in each directory are on separate disks.

Here's a closer look at the benefits of subdirectories and the workings of the directory structure.

Appreciating the Benefits of Multiple Directories

You can keep all your files in the root directory, as has been done so far in this guide's examples. If you work with low-capacity diskettes or have only a few files, you may want to continue to do just that. However, from at least two points of view—the operating system's and your own—multiple directories offer several advantages.

From the viewpoint of MS-DOS, having several subdirectories connected to the root directory helps the operating system's performance in the following ways:

- **It finds files faster.** To locate a file, MS-DOS must scan the directory until it finds the one you or your application want. The more files a directory contains, the longer it takes the operating system to find *this* file. Subdirectories make it possible for each directory to contain fewer entries, so the operating system can work faster.
- **Disks can hold more files.** A hard disk's root directory (or any single directory) can contain a limited number of entries—a limit that you might reach before the storage capacity of the disk is exhausted. When you create multiple subdirectories, however, the number of files per disk becomes almost limitless.

From your point of view, these features translate into the following benefits:

- **Directories are less confusing.** If you've ever watched the DIR command scroll dozens of names past you before you can halt the scrolling, you'll appreciate that a short list is easier to work with than a long one. With short directories, you can find file references more quickly.
- **You can keep files in logical groups.** With subdirectories, you can keep all the files belonging to an application in one directory.

Learning How Directories Are Structured

If the file drawers in your office have section dividers to separate categories of files, you may notice that the front of a drawer often contains miscellaneous files that you frequently pull. The dividers in the rest of the drawer identify categories for the remaining files.

Having all your MS-DOS files in the root directory is like having all your office files in the same drawer, but with no division into categories. The approach can work, but a system of subdirectories for different categories of files helps it work better. To see how, look at a sample directory tree.

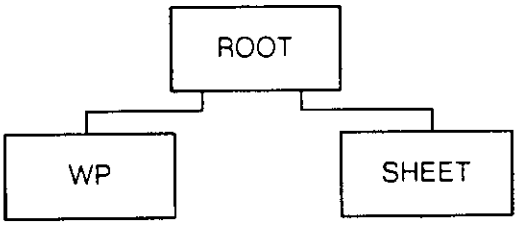
The First Level

The box below represents the root directory of drive C. It has no name; so for now, call it simply ROOT.

The root directory must contain the MS-DOS files needed for booting the computer system, along with any system configuration files (discussed in Chapter 6) and batch files (discussed in Appendix B) you use. In addition, the root directory may contain other MS-DOS files, application files, and names of second-level subdirectories.

The Second Level

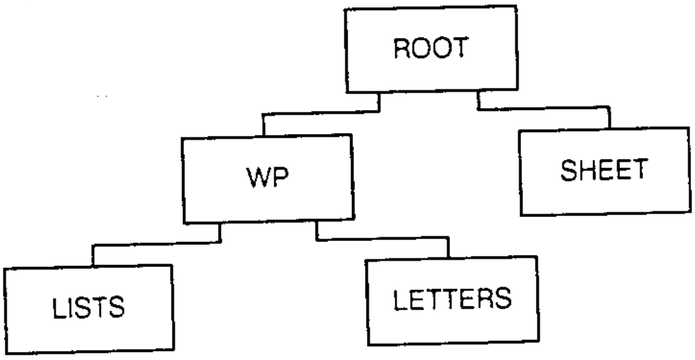
Suppose that you have two applications to run, and you want a subdirectory for each. One goes to a word-processing application (the WP directory) and the other to a spreadsheet application (the SHEET directory). The structure extends, like this:



Each second-level directory contains several files, including the program and data files for the application. Each also contains entries for any subdirectories below it.

The Third Level

Now suppose you want the WP directory to branch to two other directories: one containing data files for client, mailing, and prospective customer lists (the LISTS subdirectory) and the other containing company letters (the LETTERS subdirectory). The directory structure now looks like this:



You can continue branching subdirectories to as many levels as you want.

Creating a Directory Structure

MS-DOS automatically creates the root directory when you format the disk; you create all the others. The command you use to create a directory is MKDIR or MD ("make directory"). This section explains how to create a directory, and shows you what a root directory containing subdirectories looks like.

These instructions assume that you are creating directories on the hard disk, and that drive C is the current drive.

Making Directories

To create a subdirectory from the root directory of drive C, enter the MKDIR (or MD) command followed by the subdirectory name. The rules for naming subdirectories are the same as for naming files. For example, to create the second-level directory named WP, which was shown in the previous diagrams, enter the command:

```
C>MD \WP
```

The operating system creates the directory named WP and returns to the C> prompt. The initial backslash (\) in the directory name indicates the root directory, and says explicitly that WP is a second-level directory. Because you enter this command while the root directory is the current directory, however, you can write it this way as well:

```
C>MD WP
```

When you omit the initial backslash from a directory name, as in this example, MS-DOS assumes the directory you name is a subdirectory of the current directory.

Now that you've created the WP second-level directory, you can now create other directories or begin using this new directory.

To create a directory on a disk that is not in the current drive, include the drive designator before the directory name. With drive C as the current drive, for example, to create the WP directory on drive D you could enter the command:

```
C>MD D:\WP
```

You can create third-level subdirectories like LISTS and LETTERS in the earlier example, enter the MD command followed by a backslash (to indicate the root directory) the second-level directory name (WP), another backslash (to seperate the names of different directory levels), and the third-level directory name (LISTS). Here is the complete command:

```
A>MD \WP\LISTS
```

Because directory names follow the same rules as file names, you can create directories with file extensions. For example, you may want to create two directories with identical names but different extensions, such as SHEET.EST for worksheets containing job estimates, and SHEET.ACT for worksheets containing actual job results. Usually, though, you'll find it more convenient to keep directory names short.

Displaying the Root Directory

Now enter the DIR command to display the directory listing for the root directory of the current drive. It should look something like this:

```
Volume in drive C is WORDPROCESS
Directory of C:\

AUTOEXEC  BAT     256     6-04-87    3:12p
CONFIG    SYS     128     6-04-87    1:27p
COMMAND   COM   23617     3-21-86   12:00p
MODE      EXE   13928     7-07-86   12:00p
```

```
PRINT     EXE     8824     3-21-86   12:00p
MORE      COM      282     3-21-86   12:00p
WP           <DIR>         6-04-87   11:13a
VDISK     SYS     4256     6-11-86    3:38p
SORT      EXE     1898     3-21-86   12:00p
            9 File(s) 16247168 bytes free
```

The backslash after the drive designator in this listing means the *root directory*. The listing identifies WP as a directory, and indicates the date and time you created it. Notice that the root directory's listing doesn't include the LISTS directory because it is more than one level below the root directory.

Specifying the Directory Path

MS-DOS treats each directory as if it were a separate disk. You must access each directory to use the files and external commands listed in it. The *current directory* is the directory in which MS-DOS will try to access files; you must specify somehow the location of a file if it is not in the current directory.

When you enter a command in the current directory, the command applies only to files within that directory, unless you specify otherwise. For example, if the root directory is the current directory, the DIR command lists only files that are in the root directory. To list the contents of another directory, you must specify that directory.

To find a lower-level directory, MS-DOS looks for its name in the current directory. MS-DOS doesn't automatically know how to get there, however. When you enter a command to access a file in a subdirectory, you have to tell the operating system exactly which path to follow. This section explains how.

Writing Path Names

A *path name* is the list of directories and subdirectories that tell MS-DOS where to locate a file.

A path name has two types of elements: names of directory levels and backslashes (\). The directory names identify each level you pass through to reach the desired directory in the directory tree. Backslashes separate the names of different directory levels. (A backslash that preceeds the first directory-level name represents the root directory.) The following examples of path names refer to the directories illustrated in the earlier examples.

Path Name	Interpretation
C:\	The root directory— the first directory level.(The iniatial backslash indicates the root directory.)
C:\WP	The WP directory—second directory level.
C:\WP\LISTS	The LISTS subdirectory within the WP directory—third directory level.
C:\WP\LISTS\CLIENT.LST	A file in the LISTS subdirectory. If the last name is not a directory name, MS-DOS assumes it's a file name.

Files in separate directories can have the same name, whether or not their contents are identical. For example, these two path names:

```
C:\WP\LETTERS\CLIENT.LST
C:\WP\LISTS\CLIENT.LST
```

identify different files; even though the file names are identical, each file is in a different directory. When you enter a command, be careful to specify the correct path name or else MS-DOS may retrieve the wrong file.

Displaying the Directory Structure

The TREE command displays the directory tree structure for an entire disk. It lists each directory at each level, along with the subdirectories that branch from it. You can also ask this command to list the files contained in each directory.

☐ **Note** TREE is an external command. To run it, the TREE.COM file must be accessible (e.g., in the current directory).

To display the structure of out sample set of directories, enter the TREE command like this:

```
C>TREE
```

This command displays a hierarchical listing of all the directories and directory paths on the disk. If you want the listing to include the names of the files in the tree, add the /F option, like this:

```
C>TREE /F
```

If your disk or diskette has many files in each directory, you may want to print the listing. Turn on the printer, press Ctrl Print Screen to have the printer start echoing the screen contents, and enter the TREE command with the /F option. (After the listing is completed, press Ctrl Print Screen again to turn off printer echoing.)

Changing from One Directory to Another

When you change from the current disk drive to another drive, you enter the drive designator at the system prompt. Similarly, to change from the current directory to another directory, you use the CHDIR or CD ("change directory") command followed by the path name for the directory you want. This directory becomes the new current directory; you no longer have to specify a directory path to identify files in the new current directory.

This section explains the procedures for changing from one directory to another or from a subdirectory to another disk drive, and back again. The section also describes how to use the CD command to find your position in the directory tree, and discusses the use of directory markers.

Changing from the Root Directory to a Subdirectory

Suppose you are in the root directory of drive C. To change to the WP subdirectory, enter the command:

```
C>CD \WP
```

The operating system responds to the command by displaying the C> prompt. To verify that the current directory is indeed the WP directory, enter the DIR command; the directory name and the volume label will be displayed at the top of the list. (Later in this section, you'll learn a quicker way to verify the directory change.)

☐ **Note** You may find it helpful to change the system prompt to display the name of the current directory. For details, read about the PROMPT command in Chapter 6, "Modifying the Operating Environment."

From the WP second-level directory you can move to the LISTS directory at the third level with this command:

```
C>CD LISTS
```

Notice that no backslash was included in the directory name. Without an initial backslash, MS-DOS interprets a directory name to mean a subdirectory of the current directory. In this context, with WP being the current directory, the above command has the same result as this more explicit CD command:

```
C>CD \WP\LISTS
```

This command would take you into the LISTS directory regardless of the current directory you started from.

Changing from a Subdirectory to the Root Directory

To change to the root directory from a subdirectory at any level, enter the CD command like this:

```
C>CD \
```

Verify the change by entering the DIR command.

Changing from a Subdirectory to Another Disk Drive

From any directory on the current drive, you can change to another disk drive by simply entering the drive designator at the system prompt. Assume that you are in the WP subdirectory on drive C. To change to drive A, enter:

```
C>A:
```

Now change back to drive C from drive A:

```
A>C:
```

Although the system prompt may look the same as the C drive's root-directory prompt, MS-DOS has actually placed you back in the C:\WP subdirectory—the one you were last in on that drive.

Finding Your Position in the Directory Tree

Up to now, you've been using the DIR command to verify your position in the directory tree. A quicker way to find your position is to enter the CD command without a directory name. Suppose you are working in the LISTS subdirectory. You can display the path name to your position by entering the command:

```
C>CD
```

The operating system responds by displaying:

```
C:\WP\LISTS
```

It then returns to the system prompt.

Writing Commands with Directory Markers

When you enter the DIR command at a subdirectory level, the operating system displays a directory listing that begins like this:

```
Volume in drive C is WORDPROCESS
Directory of C:\WP\LISTS

.            <DIR>            1-12-87    1:42p
..           <DIR>            1-12-87    1:42p
CLIENT     LST      2048      2-03-87    3:51p
```

The first two entries in a subdirectory's listing show periods in place of a directory name. These periods are *directory markers*, and signify the relationship of the current directory to the directory immediately above it in the directory tree.

The one-period marker refers to the current directory. The two-period marker refers to the directory from which the current directory branches—that is, to its *parent directory*, the one immediately above it in the structure.

For example, suppose you are working in the LISTS directory. At that level, the two-period marker refers to the WP directory as the parent directory.

The two-period marker serves as a convenient shorthand when you want to change to the next higher directory and the path name is long. You can enter the CD command this way:

```
C>CD ..
```

The operating system changes from the LISTS directory to the WP directory.

Directory markers are for subdirectories only. The root directory will not contain them, since it has no parent directory.

Removing a Directory

Sometimes you no longer need a particular directory; for example, you may decide to combine the files of one directory with those of another and remove the first directory from the directory tree. The RMDIR (or RD) command enables you to remove a directory.

Removing a directory file isn't quite the same as deleting an ordinary file. You can't remove a directory if it contains any files or subdirectories.

Preparing to Remove a Directory

To prepare a directory for removal (for example, the WP directory in the earlier directory diagram), follow these steps:

1. Start at the lowest level of the directory path (i.e., the \WP\LETTERS directory in our earlier example). If this directory no longer contains files, you can remove it immediately with an RD command (see the next section). If it contains files, go to step 2.
2. Clear the LETTERS directory of files. Copy any files you want to save into other directories, then delete all files from this directory.

☐ **Note** The DEL *.* command clears all files from a directory except the directory marker files (named . and ..).

3. When the LETTERS directory is cleared of files, you can remove it with the RD command, which is explained below.
4. Repeat steps 2 and 3 for other lower-level directories (e.g., the \WP\LISTS directory), until no subdirectories or files remain in the highest-level directory you want to remove.

Now be sure you're not in the directory you're trying to remove. If you are, change to another directory.

Entering the RD Command

To remove a directory, enter the RMDIR or RD command followed by the complete path name to the directory you want to remove. For example, to remove the LETTERS directory, enter this command:

```
C>RD \WP\LETTERS
```

If the directory is already empty and you specified a valid directory path, the operating system will remove the directory and return to the system prompt.

However, if the directory you specify still contains files, hidden files, or subdirectories (or if you enter this command when the directory you're trying to remove is the current directory), the operating system will display an error message, cancel the command, and return to the system prompt.

To remove a directory from a disk that is not on the current drive, include the drive designator as part of the path name. For example, if drive A is the current drive and you want to remove the LETTERS directory from drive C, enter the command:

```
A>RD C:\WP\LETTERS
```

The operating system removes the LETTERS directory from drive C and returns to the system prompt.

5

Managing Files on Disk

This chapter is the last of three on file management. Here, the focus is on disk operations: copying files to other disk locations, and backing up and restoring the contents of a hard disk.

Learning More about Copying Files

In Chapter 1, you saw how to copy the contents of an entire diskette by using the DISKCOPY or the COPY *.* command. This section summarizes how to copy individual files to various target locations and how to combine files using COPY. It also explains how to copy directories and subdirectories with the XCOPY command. Finally, the section compares the features of the COPY, XCOPY, and DISKCOPY commands.

Copying Files to Alternate Locations

The following table summarizes various ways to use COPY to transfer the contents of one or more files to alternate disk locations. As you read the table, remember that MS-DOS requires you to change the file name or extension if you're copying to another location on the same disk or in the same directory. All sample entries assume the root directory of drive C as the current directory.

Sample Entry	Source Location	Target Location
COPY CLIENT.LST MAIL.LST	One file in current drive and directory.	Different filename in same drive and directory.
COPY CLIENT*.* MAIL*.*	Multiple files with related file names in current drive and directory.	Different related file names in same drive and directory.
COPY CLIENT.LST A:	One file in current drive or directory.	Same file name in root directory of another drive.

Sample Entry	Source Location	Target Location
COPY A:CLIENT.LST	One file in root directory of a noncurrent drive.	Same file name in root directory of the current drive.
COPY B:MAIL.LST A:\WP\LISTS\MAIL87.LST	One file in root directory of a noncurrent drive.	Different file name in subdirectory of a noncurrent drive.
COPY A:\WP\CLIENT.LST B:\ADDRESS	One file in subdirectory of a noncurrent drive.	Same file name in a subdirectory of a second noncurrent drive.

The COPY command has many options you may be interested in after you have more MS-DOS experience. For information about these options, see the online HELP message about the COPY command.

Combining Files

The COPY command also lets you combine several source files into one target file. In this procedure, MS-DOS copies the first source file named in the command to the target file; then the second file, which it appends to the end of the first in the target file; then the third, which it appends to the end of the second; and so on, until the system copies and appends all the files named. You can combine as many source files as you want.

To enter the command, separate the source file names with plus signs (+). The target file is always the last file named in the command. (Do not put a plus sign between the last source file name and the target file name.) Here are some ways to combine files. (In these sample entries, assume the root directory of drive C is the source directory.)

Sample Entry	Source Location	Target Location
COPY CLIENT1.LST + CLIENT2.LST + CLIENT3.LST MAIL.LST	Three files, all in current directory and drive.	One file in same directory and drive.
COPY CLIENT*.LST MAIL.LST	Multiple files with same extension and similar file names in current directory and drive.	One file in same directory and drive.

Sample Entry	Source Location	Target Location
COPY CLIENT1.LST + B:\WP\CLIENT2.LST + \WEST\CLIENT3.LST A:MAIL.LST	First file is in current drive and current directory; the second file is in a subdirectory on drive B; the third file is in noncurrent directory of current drive.	One file in root directory of drive A.
COPY *.LST + *.SRT MAIL.ALL	Multiple files with two different extensions in current directory and drive.	One file in same directory and drive.

Using wild-card characters to combine files has its dangers. For example, if you use a wild card to tell MS-DOS to copy all files with an .LST extension to a file called MAIL.LST—and if one of the files to be copied is already called MAIL.LST—the operating system overwrites the original MAIL.LST when it copies the first .LST file it finds. By the time MS-DOS tries to find the original MAIL.LST, it has already erased the file. Thus, a command like the following may look innocent to you:

```
A>COPY *.LST MAIL.LST
```

However, to MS-DOS it's an order to delete a file. You'll see an error message telling you that a file that was to have been copied is lost. To avoid this problem, always specify a target file name that's different from the names of any of the source files.

Copying Entire Directories and Subdirectories

The COPY *.* command (described earlier) copies all files within the source directory specified in the command. If the disk has only one directory, COPY *.* copies all files on the disk. If the disk has multiple directories, however, COPY *.* only copies files from the specified directory.

Sometimes, you want to copy files in the current directory and all its subdirectories. In that case, the most efficient way is to execute the XCOPY ("extended COPY") command.

XCOPY works just like COPY, except that it assumes you want to copy all files in a source directory unless you specify a path name and a file name. For example, to copy all the root directory files from drive C to drive A, enter the command:

```
C>XCOPY C: A:
```

This command is the same as entering **COPY C:*.* A:**. The following table summarizes the special features of XCOPY:

If You Want to Copy	Sample Entry
Current directory files created on or after a specified date.	XCOPY C: A: /D:1-15-87
All current directory files plus all files from its subdirectories (does not copy empty subdirectories).	XCOPY C: A: /S
All current directory files and all subdirectories, including empty subdirectories (the /S must always be included in the command when you specify /E).	XCOPY C: A: /S /E

For information about all XCOPY options, enter the command **HELP XCOPY**.

Deciding Which Copying Command to Enter

As you saw in Chapter 1, the COPY *.* and DISKCOPY commands both copy entire diskettes; so does the XCOPY command. However, the three commands have different features, which are compared in the following table.

COPY	XCOPY	DISKCOPY
Copies files to or from diskettes and hard disks	Copies files to or from diskettes and hard disks	Works with diskettes only; doesn't copy to or from hard disks
Copies all files from the current directory, one whole file at a time, regardless of how the files are arranged on the source disk	Copies all files and directories one whole file at a time, regardless of how the files are arranged on the source disk	Creates an exact duplicate of the source diskette, copying files and directories onto the target diskette exactly as they appear on the source diskette
Requires that the target disk be formatted before copying	Requires that the target disk be formatted before copying	Automatically formats the target diskette with the same number of tracks and sectors as the source diskette

COPY	XCOPY	DISKCOPY
Copies files regardless of the date created or updated	Lets you copy files created or updated on or after a specified date	Copies the diskette regardless of the date of file creation or updating
Doesn't copy subdirectories	Can copy subdirectories if you include /S in the command	Always copies subdirectories
Doesn't copy MS-DOS hidden system files (you must do that with the SYS or FORMAT command)	Doesn't copy MS-DOS hidden system files (you must do that with the SYS or FORMAT command)	Always copies MS-DOS hidden system files if they exist on the source diskette

DISKCOPY is good for copying diskettes that contain MS-DOS system files. COPY and XCOPY are better if the source disk has undergone heavy use; if files are fragmented on the source disk, they will be contiguous on the copy made with these commands. In any case, only COPY or XCOPY (not DISKCOPY) can be used if you're copying to or from hard disks.

Backing Up and Restoring Your Hard Disk

The term *backing up* means copying information from a working disk to a diskette that you'll store to protect the information. The opposite process is *restoring*, which means copying the information from the reserve diskette back to a working disk.

The information on your disks is very valuable and costly to replace. Because you can't always predict when events like power surges or failures may destroy information, you should establish a routine of regular and frequent hard-disk backups.

One way to back up a hard disk is to equip your system with a streaming tape drive. MS-DOS, however, gives you a way to back up a hard disk onto diskettes with the BACKUP and RESTORE commands.

These two commands always work together. If you use BACKUP to copy information from a hard disk, you must use RESTORE to copy it back from diskettes. (You cannot make normal use of a file you've backed up with BACKUP until you restore it with RESTORE.) These commands can copy and restore specific files, groups of files, or the contents of an entire hard disk (including the MS-DOS system files).

Caution If the disks you use include copy-protected software, consult your software instruction manuals about backup procedures. BACKUP and RESTORE may make some copy-protected software unusable.

Caution If you've used the JOIN or SUBST command, don't use BACKUP as long as these commands remain in effect. You may damage or lose files. Cancel these two commands before you proceed with the backup operation.

Backing Up an Entire Hard Disk

Before you begin a backup operation, be sure to format enough diskettes to complete the task. When you are backing up a hard disk, you should have a few more diskettes than it takes to hold the data now on the hard disk.

To estimate the number of formatted diskettes you'll need, run the CHKDSK command to find how many bytes are required by files on the hard disk. Divide this amount by the capacity of a diskette. For example, if you're backing up a nearly full 20MB hard disk onto 360K diskettes, you may need as many as 56 diskettes.

With your MS-DOS system in the current drive, enter the BACKUP command followed by the source drive designator, the target drive designator, and the /S option to back up all directories (if the source drive is the current drive, you can omit the drive designator). For example, if you are backing up files from drive C onto drive A, enter the command:

```
C>BACKUP C: A: /S
```

MS-DOS prompts you to insert backup diskette 01 into drive A. Insert the first target diskette and press any key.

The system writes a backup ID number on the target diskette—in this case, 01—and begins copying files onto it. If the backup diskette fills up before the operation ends, the system prompts you for backup diskette 02. In that case, remove the first backup diskette from drive A and label it with the date and the disk number (for example, "Backup 01"). Then insert the second backup diskette and press a key. Repeat this procedure until MS-DOS returns to the system prompt.

To back up only those files changed on or after a specified date, include the /D: option and the date in the command. This provides an efficient way to keep an existing set of backup diskettes up to date. Enter a command such as this:

```
C>BACKUP C: A: /S /D:1-15-87
```

(Only hyphens, not slashes, are permitted within the date you type in the BACKUP command.)

Backing Up Individual Files and File Groups

To back up an individual file, you modify the previous procedures by specifying the name of the file you want to back up. For example, to back up a file named CLIENT.LST from drive C to a diskette in drive A, enter:

```
C>BACKUP CLIENT.LST A:
```

The operating system writes the backup ID number on the diskette, copies the file, and returns to the system prompt.

To back up a group of files, use wild-card characters in the command. For example, to back up all .LST files, you can enter the command this way:

```
C>BACKUP *.LST A:
```

☐ **Note** These examples assume that you are beginning a new backup diskette. (If you are adding a file to an existing backup diskette, see "Adding Files to Existing Backup Diskettes.")

Adding Files to Existing Backup Diskettes

To back up a file onto a diskette already created with the BACKUP command, use the /A option. For example, to add CLIENT.LST to the files already stored on a backup diskette, enter the command:

```
C>BACKUP CLIENT.LST A: /A
```

Insert the existing backup diskette into drive A and press any key to begin the operation. The system copies the file and returns to the system prompt.

Restoring Files

As explained earlier, *restoring* a file means copying it from a backup diskette onto a working hard disk. The RESTORE command only works with diskettes created by the BACKUP command, not those created by COPY, XCOPY, or DISKCOPY.

The RESTORE procedure is the reverse of the BACKUP procedure. If you previously backed up information onto two or more diskettes, you should have labeled the backup diskettes with sequence numbers. To restore the information, you insert the diskettes in the same order in which you backed them up.

Here is the procedure for restoring all files to the hard disk:

1. With your MS-DOS user files in the current drive, enter the RESTORE command followed by the source drive designator and the target drive designator (if the target drive is the current drive, you can omit its drive designator). For example:

   ```
   C>RESTORE A: C: /S
   ```

   will restore all the files from diskettes in drive A into all directories on drive C:

2. As prompted, insert the first backup diskette into drive A and press any key to begin.

   MS-DOS displays the file name as it restores each file, until it reaches the end of the diskette. If the earlier backup procedure required more than one diskette, the operating system prompts you to insert the next diskette.

3. The restore procedure ends when MS-DOS returns to the system prompt. To verify that all files are restored, use the DIR command to list the directory of the target disk.

You can restore an individual file by including the name of the file you want to restore after the source drive designator and the target drive designator (if specified). To restore a group of files, you can use wild-card characters in the command.

After you finish restoring your files, you can put away the backup diskettes until the next backup operation.

Establishing a Periodic Backup Routine

A carefully controlled rotation of backup diskettes helps ensure that data loss from any cause will be minimal. For example, you can back up your hard disk daily on three sets of diskettes.

- Day 1, back up the hard disk onto the first set of diskettes.
- Day 2, back up onto the second set.
- Day 3, back up onto the third set.
- Day 4, reuse the first set of diskettes to make your backup.

You'll always have the data preserved in the state it was on the previous two days.

6

Modifying the Operating Environment

The term *operating environment* refers to the internal conditions that determine the way MS-DOS works—how the operating system accepts input and produces output.

So far, you've been using the MS-DOS default environment —that is, the operating conditions already set up when you boot the system. Some MS-DOS commands modify aspects of the default environment. This chapter explains how to change the MS-DOS system prompt, how to redirect output from one device to another, and how to modify your default system configuration.

Changing the System Prompt

Up to now, this guide has used the default version of the MS-DOS system prompt—the drive designator followed by a right-pointing angle bracket (for example, A>). The PROMPT command enables you to change this prompt. It is an internal command, so you can enter it from any directory.

If you create and use subdirectories, you'll probably find it convenient to see the current directory reflected in the system prompt. You can change the standard system prompt to include the current directory path by entering the PROMPT command this way:

```
A>PROMPT $P
```

The dollar sign ($) in the command tells the operating system that the following character is a special code for standard prompt text. The character P is the code signifying the current directory path.

When you press **Enter**, the operating system replaces the old prompt with the new one:

```
A:\
```

This prompt indicates that you are in the root directory. If you change to a subdirectory like SHEET, the prompt would look like this:

```
A:\SHEET
```

With the PROMPT command, you can vary the content of the system prompt in several ways. You can use the command to display a message. Your PROMPT command can also include combinations of the $ symbol and a code character that cause the prompt to include special symbols (such as the right angle bracket) and variable information (such as the current directory, date, and time). The table below shows some possibilities. (In all the examples of the resulting prompts, A is assumed to be the active drive.)

Text or Code	What It Displays	Sample Entry	Result
<text>	The text entered	`PROMPT Enter a command:`	`Enter a command:`
$N	The current disk drive	`PROMPT $N`	`A`
$D $T	The date and time, separated by a space	`PROMPT $D $T`	`Thu 1-22-1987 13:30:57.50`
$P	The current directory path	`PROMPT $P`	`A:\`
PG	The current directory path and a right angle bracket	`PROMPT $P$G`	`A:\>`
P_<text>	The current directory path, a line break, and text	`PROMPT $P$_Enter a command:`	`A:\` `Enter a command:`
$S	No prompt (cursor only)	`PROMPT $S`	`<cursor>`
$S $G	A right angle bracket in the fifth column of the screen (i.e., four spaces are between $S and $G)	`PROMPT $S    $G`	`    >`

Since you can change the prompt whenever you want, you can freely experiment with different combinations of text and prompt codes. To change any prompt back to the default system prompt (which is the current drive designator and a right angle bracket), enter the command

```
PROMPT
```

without any code characters or text following the PROMPT command.

☐ **Note** The changed prompt remains in effect until you reboot the operating system or turn off the computer. To display your preferred system prompt each time you boot MS-DOS, include the PROMPT command that creates it in an AUTOEXEC.BAT batch file (see Appendix B, "Batch Files").

Redirecting Output

When you enter a command that results in data being output (e.g., a directory listing), the operating system normally directs that data to the video monitor. This section explains how you can redirect this output to a printer or a file instead, and how you can control output by sorting it or displaying it one screenful at a time.

This section also shows you how to use the MODE command to set the characteristics of a serial printer and to redirect printed output from the default parallel printer port to the serial printer port instead.

Sending a Report to the Printer

On most computer systems, the *standard output device* is a video monitor. When you enter commands like DIR or TYPE, the operating system sends the resulting data output to the monitor. However, you can bypass the video display altogether and send the output directly to the printer. To do so, type the command that causes the report, followed by the "output redirect" symbol (the > character) and the name of the port to which the printer is attached.

The following example assumes that your printer is attached to the PRN port (i.e., the default parallel printer port). You'll print the root directory of drive A.

To start, be sure your printer is on. Then enter the DIR command like this:

```
A>DIR > PRN
```

Printing begins immediately and the directory listing is printed as it would appear on the screen. The system prompt reappears when printing ends.

You can also use this feature to redirect the output to another device (identified with one of the standard device names described in Chapter 2) or to a file. For example, if you want to store a copy of the directory listing in a file on drive A called DIR.TXT, enter the command:

```
A>DIR > DIR.TXT
```

If the file doesn't exist, MS-DOS creates it and stores the directory there. If a file of that name already exists, MS-DOS replaces the existing file with the new one. The system prompt reappears when the operation ends.

Rearranging Command Output

MS-DOS includes commands that are called *filters*. A filter command rearranges the output of another command before sending the output to an output device. For MS-DOS to recognize a filter command, you must precede the command name with the vertical bar character (¦).

One filter command is SORT, which takes list-style output (such as the directory listing displayed by the DIR command) into alphabetical or numerical order. If you want to send a sorted version of the directory listing to a file called DIR.SRT on drive A, enter the command:

```
A>DIR ¦ SORT > DIR.SRT
```

The operating system creates the directory listing, sorts it in alphabetical order, creates the file DIR.SRT on drive A, and sends the sorted directory listing into the new file.

The MORE filter command enables you to display output one screen at a time. For example, if you want to display the contents of a long letter (called SMITH.LTR) screen by screen, enter the command:

```
A>TYPE SMITH.LTR ¦ MORE
```

MS-DOS displays the first 23 lines of the letter and pauses; the message `-- MORE --` appears at the bottom of the screen. Press any key to display the next 23 lines.

These filter commands are external commands, which means the MORE.COM or SORT.EXE files must be available on a drive when you include MORE or SORT with another command.

Configuring for a Serial Printer

Say you have just connected a serial printer to the first serial port (COM1:). Now you must tell MS-DOS what this serial printer's characteristics are and arrange for the operating system to direct output to its port. To perform these functions, you use the MODE command.

☐ **Note** Your application program may require that you set the system with the MODE command in a particular way to be able to print output from the program. Check the application program's manual to determine the proper MODE command.

Defining the Serial Printer

You must tell MS-DOS the characteristics of the serial printer before it can communicate with the device. These characteristics include *baud rate* (printer speed in bits per second), *parity* (error checking during printing), *data bits* (character definition), and *stop bits* (an end-of-character marker).

☐ **Note** Refer to the manual for your serial printer to find the proper values for these characteristics and to the HELP program for information on the MODE command and an explanation of its parameters.

For example, to assign the first serial port to a printer that operates at 9600 baud with no parity, eight data bits, and one stop bit, you can enter this command:

```
A>MODE COM1:9600,N,8,1,P
```

The final parameter (P) calls for your computer to make continuous attempts to communicate with the printer after it receives a time-out error signal (e.g., printer off line) from the printer. Always include this parameter when installing a serial printer.

Redirecting Output to the Serial Port

Even though you have attached a serial printer to the computer and have defined the characteristics of the COM1: port, MS-DOS will still try to use the first parallel printer port (LPT1:) for some printing operations unless you tell it otherwise. Therefore, the next step in setting up the serial printer is to redirect output intended for LPT1: to COM1: instead.

You accomplish this with a different form of the MODE command, which has the name of the port you're redirecting output from, an equal sign (=), and the name of the port you want the output sent to. For example:

```
A>MODE LPT1:=COM1:
```

will direct output to the first serial printer port whenever you enter a command that would have printed output through the first parallel port.

To change this port reassignment back to the original setting, enter the MODE command this way:

```
A>MODE LPT1:
```

and the operating system will display a message that says LPT1: is not rerouted. (In other words, the operating system resumes its original port assignment.)

Combining MODE Commands

The MODE command can perform several other functions besides the two described here (enter the command **HELP MODE** for more information). If you need to enter several MODE commands to set up your computer as you desire, MS-DOS allows you to do so without each command cancelling the effects of the other commands. For example, if you need to define a printer's characteristics and also change its port assignment, you can enter these two command lines individually:

```
A>MODE COM1:9600,N,8,1,P
A>MODE LPT1:=COM1:
```

MS-DOS resumes its default modes when you reboot or turn off the computer. If you want to have this printer configuration everytime you use the computer you may want to include the appropriate MODE commands in a batch command file (see Appendix B, "Batch Files").

Modifying Your System Configuration

The system configuration is an aspect of the operating environment that governs *how* the system does something, rather than *what* it does. Your computer system already has a default system configuration. If you want a different system configuration, MS-DOS allows you to create a file called CONFIG.SYS that can contain a list of your commands that redefine the default configuration of the operating system.

You must create a CONFIG.SYS file and write the proper commands in it to activate some of the operating system's features. For example, if you want to activate the virtual disk drives or expanded memory handling features of Enhanced MS-DOS (which are explained in Chapter 7), you first must understand how to set up a CONFIG.SYS file.

Understanding Configuration Commands

System configuration commands differ from the MS-DOS commands you've seen so far. You don't enter configuration commands directly from the keyboard. Instead, you put them into a system definition file, called the CONFIG.SYS file. When you boot MS-DOS, the operating system checks the CONFIG.SYS file and executes the commands in it. Thus, the file must be in the root directory of the drive or diskette from which you boot the operating system.

Each configuration command you write in a CONFIG.SYS file will look like a mathematical equation. You enter the command name, an equal sign (=), and then the desired value.

Displaying the Current Configuration

You can find out if your operating system has a CONFIG.SYS file already installed by listing the root directory. If a file with that name exists, you can display its contents by entering the TYPE command as follows:

```
A>TYPE CONFIG.SYS
```

The operating system displays the file and returns to the A> prompt. You can now see the current configuration of your system.

☐ **Note** If you already have a CONFIG.SYS, creating a new CONFIG.SYS file automatically erases the existing file. To avoid destroying the current configuration information, copy the existing CONFIG.SYS file to another disk before you practice entering configuration commands.

Creating the CONFIG.SYS File

You can create a CONFIG.SYS file in the root directory using a word-processing or text editor program. The method described here, though, uses the COPY command to cause a short file you write at the console (the keyboard and screen) to be saved under the file name CONFIG.SYS.

Creating a File at the Console

To begin creating the file, enter this command:

```
A>COPY CON CONFIG.SYS
```

The device name CON represents the console; the COPY command can deal with device names as if they were file names. This command tells MS-DOS that the lines you will type at the console are the source it is to copy into the target file named CONFIG.SYS.

After you enter that COPY command, the cursor advances to the beginning of the next line. At this point, you can enter configuration commands. (What these commands do and how you write them will be described in the next section.) When you finish typing each line, press Enter; the cursor will drop to the next blank line.

You can edit each line with ←Backspace while you're on that line; you can't however return to change earlier lines. If you notice an uncorrectable mistake, you can cancel the COPY command by pressing Ctrl Break.

A typical CONFIG.SYS file might contain the following commands:

```
BUFFERS = 10
DEVICE = ANSI.SYS
DEVICE = VDISK.SYS 100
```

The exact set of commands, however, will depend on the configuration you want for your system.

After you enter the last configuration command to be in the CONFIG.SYS file, press Enter to move the cursor to the blank line that follows that last command. Now press Ctrl Z to insert an end-of-file symbol, which will conclude the file you've written. Press Enter and MS-DOS will display this message:

```
1 File(s) copied
```

The file called CONFIG.SYS now exists on your disk. You can examine its contents by entering this command:

```
TYPE CONFIG.SYS
```

Running the CONFIG.SYS File

After you create a new CONFIG.SYS file to reconfigure the computer system, the system's present configuration remains in effect until you activate the new configuration. To do so, you must reboot MS-DOS. Each time you boot up the computer after that, MS-DOS will execute the configuration commands in the new CONFIG.SYS file just after it loads the system files and just before it executes the commands in an AUTOEXEC.BAT file (see Appendix B for information on AUTOEXEC.BAT).

CONFIG.SYS Commands

The following table lists the system configuration commands you can include in a CONFIG.SYS file. It defines each command and provides the general form for entering it. To see all the settings, configurations, file names, or parameters that can be entered with each configuration command, refer to the command's online HELP screens.

Command	Entry Format	Explanation
BREAK	BREAK=ON or OFF	Determines the conditions during which MS-DOS will check if you have pressed Ctrl Break or Ctrl C to give a break signal.
BUFFERS	BUFFERS=*number*	Sets the number of temporary memory areas where MS-DOS holds data being read from or written to disk.
COUNTRY	COUNTRY=*code*	Uses the time, date, and currency formats for the specified country.
DEVICE	DEVICE=*driver file name*	Installs the specified device driver file. Any number of DEVICE commands can be in a CONFIG.SYS file.
DRIVPARM	DRIVPARM=*parameters*	Defines alternative settings for drive devices.
FCBS	FCBS=*nopen,nprotected*	Sets the maximum number of files that file control blocks (FCBs) can have open at once.
FILES	FILES=*number*	Sets the maximum number of files that MS-DOS can have open at the same time.
LASTDRIVE	LASTDRIVE=*drive letter*	The last drive designator that MS-DOS accepts as valid.
SHELL	SHELL=*parameters*	Loads an alternative command processor file in place of the default COMMAND.COM file.

Installing a Device Driver

One typical configuration command is DEVICE, which you use to install device drivers. A *device driver* is a program that allows the operating system to control a peripheral device, such as a disk drive. The device drivers usually are files with the .SYS file name extension.

Your Enhanced MS-DOS diskettes include several device driver files: ANSI.SYS, DRIVER.SYS, LCDPANEL.SYS, VMEM.SYS, and VDISK.SYS.

- ANSI.SYS enables your display to be controlled by programs that include codes known as *ANSI escape sequences*.
- DRIVER.SYS allows you to install an external disk drive.
- LCDPANEL.SYS enables you to direct output to an LCD front panel (if your computer has one) through the device name LCDDEV (see Chapter 7).
- VMEM.SYS allows you to use your computer's extended memory (i.e., RAM at addresses above 1MB) with software that takes advantage of the LIM/EMS and EEMS standards for expanded memory (see Chapter 7).
- VDISK.SYS allows you to reserve a specific amount of RAM for use as a disk drive. A "drive" you create in RAM this way is called a *virtual disk*.

  One advantage of a virtual disk is that files and programs can be accessed from it much faster than from diskettes. A virtual disk can be most useful if you copy frequently used programs, commands, and data files into it.

☐ **Note** The VDISK.SYS configuration stays in effect until you reboot the operating system or turn off the computer. If you want to save the information on the virtual disk, be sure to copy it to a physical disk before turning off the computer.

To set up a virtual disk drive, include a DEVICE command in the CONFIG.SYS file. The default version of this command is:

```
DEVICE=VDISK.SYS
```

This creates a virtual drive with a size of 64K made of RAM in standard memory. You can make a larger virtual disk by specifying the number of kilobytes you want it to contain. For example, this command:

```
DEVICE=VDISK.SYS 150
```

creates a virtual disk that can hold files that total 150K.

You can enter additional DEVICE commands to create more than one virtual disk. For example, if your system had one hard disk (drive C), by including these two lines in the CONFIG.SYS file

```
DEVICE=VDISK.SYS 150
DEVICE=VDISK.SYS
```

you create two virtual disks: the first with 150K is known as drive D, the other is drive E and has the default 64K size. VDISK will display the designator names of virtual disks when you boot up the computer.

If your computer contains extended memory (i.e., RAM at addresses above 1MB), you can treat any of that memory as a virtual disk. To do so, your command must include VDISK's /E parameter. For example, this command:

```
DEVICE=VDISK.SYS /E
```

creates a 64K virtual disk in extended memory. If you want to make a larger virtual disk in extended memory, your command must include three additional pieces of information: the disk size, the sector size (usually 128 bytes), and the maximum number of files that can be written to the virtual disk (usually 64 files). To create a 360K virtual disk in extended memory, you can enter this configuration command:

```
DEVICE=VDISK.SYS 360 128 64 /E
```

You can use the VDISK, VMEM, and CACHE features of Enhanced MS-DOS simultaneously in extended memory without conflict. See Chapter 7 for information about using VDISK in conjunction with the other two features.

7

Enhanced MS-DOS Special Features

Enhanced MS-DOS includes special features and commands that are not available in other versions of the operating system. If you are already familiar with another version of MS-DOS, this chapter will help you understand quickly how to use these additional capabilities.

The enhanced features include: support for disks over 32MB through the FDISK and FORMAT commands, the CACHE command, LIM (Lotus-Intel-Microsoft) expanded memory emulation through the VMEM.SYS device driver and VMEM command, the DISKPACK command, the SPEED command, the LCDPANEL command, and the LOCK command. This chapter provides an in-depth discussion of each of these features.

Taking Advantage of Large-Disk Support

Many earlier versions of MS-DOS supported only hard disks of 32MB or less. This enhanced version also supports "large disks"—that is, disks larger than 32MB. It also allows you to divide a hard disk of any size into several partitions, each of which can be greater than 32MB.

If you have a hard-disk system, you followed the instructions in Chapter 1 and used FDISK to prepare the entire hard disk as a single MS-DOS partition. Now you'll learn what a logical disk drive is, and how to partition your hard disk into one or more logical drives.

Distinguishing Physical and Logical Disk Drives

In your installation and assembly manual and in this user's guide, you've worked with *physical* disk-drive hardware. MS-DOS recognizes the first physical hard disk in the computer system as drive C, and the second physical hard disk as drive D. When you ran the SETUP program during installation and assembly, you identified and recorded the physical disk drives attached to your computer.

When you divide a physical disk into more than one partition with the enhanced FDISK command, each partition on the disk is called a *logical* disk drive, which MS-DOS treats as if it were an individual drive with its own designator.

Suppose you have a single hard disk of 117MB and you've divided it into three partitions. MS-DOS always recognizes the first partition on the disk as physical drive C, the second partition is logical drive D, the third is drive E, and so on.

If you later add a second physical hard disk and identify it to MS-DOS through the SETUP program, the drive designators automatically change. The second physical drive becomes drive D. The designators for the logical drives on drive C then begin with drive E, as illustrated in the following table of possible drive-designator assignments:

Hard Disk Partition	Designator on Single Physical Hard Disk Drive	Designators on Multiple Physical Hard Disk Drives	
		Drive 1	Drive 2
1	C (physical)	C (physical)	D (physical)
2	D (logical)	E (logical)	H (logical)
3	E (logical)	F (logical)	I (logical)
4	F (logical)	G (logical)	J (logical)

Because logical drive designators change according to the number of physical disk drives, be sure to read the instructions in "Changing the Drive Sequence" when you add or remove a second hard disk.

Creating Logical Drives

With the FDISK command, you can create up to four MS-DOS partitions of varying sizes on your hard disk. Whenever you create more than one partition, the operating system automatically assigns logical drive designators to the partitions.

Caution If you already have files on the hard disk, back them up before starting; FDISK deletes all information stored on the disk. If you are partitioning a new physical disk, be sure you first run the SETUP program, which is described in your computer's installation and assembly manual.

To create the logical disk drives by partitioning drive C, insert your MS-DOS diskette in drive A and enter the FDISK command. Then follow this procedure:

1 If you had previously created one or more MS-DOS partitions on the hard disk, you must eliminate them before you can establish a new combination of logical drives. From the list of FDISK options, choose the one to delete MS-DOS partitions.

2 When there are no existing partitions, choose the option to create an MS-DOS partition.

3 When FDISK asks if you want to use the entire fixed disk for the operating system, answer *no*. The display now displays the total number of cylinders on the hard disk, the maximum available space, and the beginning cylinder position for the partition.

   (If you were to respond *yes* to this question and your drive was larger than 32MB, FDISK would make the entire disk one MS-DOS partition.)

4 FDISK prompts you for the size of the first partition (i.e., the number of cylinders). If you want to create multiple partitions, enter a cylinder number proportional to the total number; for example, if you are creating four logical disk drives, enter a cylinder number that represents about 25 percent of the total number of cylinders available on the disk.

Caution A few application programs require strict IBM compatibility. If you plan to run such programs, your first partition on the physical disk drive must be 32MB or less and these programs should be copied onto that partition only.

☐ **Note** If your hard disk is larger than 32MB, the default value that appears on the screen when you're asked for the size of the first partition is the number of cylinders that will make that first partition exactly 32MB.

5 The operating system prompts you for the starting cylinder, and displays a number based on the partition size you indicated in Step 3. If you want to locate the partition at that cylinder, press Enter; otherwise, enter a different starting cylinder number.

6 To create the remaining partitions, return to the FDISK menu, select the option to create an MS-DOS partition, and repeat the above steps.

7 When you've created more than one partition on the first physical drive, you must designate one of the logical drives as the *active* (i.e., bootable) drive. Otherwise, you cannot boot up MS-DOS from the hard disk, even if the system files are present on drive C.

Return to the FDISK menu and select the option to change the active partition. The display will tell you the active status of the existing partitions, and prompt you to enter the number of the partition you want to make active.

☐ **Note** Only one logical drive on drive C can be active at a time.

8 After you've created all the partitions you want, and have made one of them active, reboot with a system diskette in drive A to establish the new partitions.

9 Now you must format each partition as if it were a separate drive. To format the active partition, enter this command (put the active logical drive's designator in place of D:):

```
A>FORMAT D: /S /V
```

Nonactive logical drives don't need to contain the system files, so you omit the /S parameter when you format those drives.

After you finish formatting the physical drive and the logical drives, you can copy files into them using the appropriate form of the RESTORE, COPY, or XCOPY command.

Changing the Drive Sequence

Whenever you install a second physical hard disk, MS-DOS automatically recognizes the new disk as physical drive D. The names of all logical drives on the first physical drive (drive C) change correspondingly (e.g., drives C, D, E, and F become drives C, E, F, and G). Therefore, you must back up all files in all partitions of the first physical drive before installing the second physical drive.

Any time you do anything that changes the logical drive configuration, such as varying the number or sequence of drives, you must run FDISK and create a new set of partitions. For example, if you intend to disconnect the second physical disk drive from your computer system, you must first back up all the files on physical drive C so you can restore them after you remove drive D and then redefine the partitions on physical drive C.

Creating a Disk Cache

The CACHE command reserves a portion of RAM that serves as a buffer (called the *cache*) that holds a copy of the information most recently read from or written to a hard disk. When a program needs to access the disk, the software will first check to see if the desired information is already contained in the cache; if so, the program accesses the information from RAM rather than reading it again from the disk. Whenever the desired information isn't in the cache, the program will read from the disk, as usual, but the new information will be added to the cache, since it's likely to be requested again. In this way, CACHE increases the hard disk's effective speed.

If your program needs to revise or add to disk information that is held in the cache, both the cache and the disk itself are revised at the same time. This means that, when you use the CACHE command, the disk is always kept up-to-date without your having to make a conscious effort to copy information held in memory onto disk.

Turning Caching On or Off

The default version of the CACHE command

```
C>CACHE ON
```

turns on caching for all hard disks and reserves 64K in standard memory for the cache buffer.

You can specify a larger buffer size with the /S: option. The following command sets the buffer size at 256K:

```
A>CACHE ON /S:256
```

You can restrict caching to a specific physical hard disk by including its drive designator with the CACHE command. For example, this command turns on caching for drive C only:

```
C>CACHE C: ON /S:150
```

Drive designators in the CACHE command can only refer to entire *physical* drives, not to individual *logical* drives.

If disk caching is on but you want to turn it off and return to normal disk read and write operations, enter this command:

```
C>CACHE OFF
```

Installing the Cache in Extended Memory

If your computer has extended memory located at addresses beyond 1MB, you can locate your cache in extended memory rather than in standard system memory (640K and lower). Including the /E parameter with the CACHE command puts the cache in extended memory. For example, this command:

```
C>CACHE ON /E
```

creates a default-sized 64K cache in extended memory. You can devote more of extended-memory to the cache by including the /S parameter:

```
C>CACHE ON /S:256 /E
```

This command creates a 256K cache in extended memory. You also can specify the starting location of the cache in extended memory with the /L parameter. This command:

```
C>CACHE ON /S:115 /E /L:256
```

places the 115K cache starting after the first 256K in extended memory.

Displaying the Cache Statistics

The CACHE command can display information about the number of disk input and output operations and the effect caching has on overall computer system performance. To display these statistics, enter this command:

```
A>CACHE STAT
```

ng EMS-Compatible ware with Extended Memory

If your system has RAM above the 1MB address (known as *extended memory*), the VMEM.SYS device driver enables you to use this memory with software that's designed to take advantage of the Expanded Memory Specification (EMS) developed by Lotus, Intel, and Microsoft (also called the *LIM specification*).

The VMEM.SYS device driver performs a software emulation that allows your system's extended memory to emulate an EMS-compatible expanded memory board. This emulation method also allows your computer's extended memory to conform to the Enhanced EMS specification (EEMS), developed by AST, Ashton-Tate, and Quadram. The main advantage of this enhanced specification is that you can devote more than four 16K blocks of standard memory (i.e., RAM in the 0 to 640K range) to expanded-memory paging. (Four contiguous 16K blocks of standard memory is the LIM specification's default and limit.)

Installing VMEM.SYS

To load the VMEM.SYS device driver, include a DEVICE command like the following in your CONFIG.SYS file:

```
DEVICE = VMEM.SYS [/L:rblocks] [/P:pblocks] [/E:eblocks]
```

The /L, /P, and /E parameters have the following meanings:

/L:*rblocks* specifies the number of 1K blocks of extended memory you wish to reserve for use by other programs before devoting any to VMEM.

/P:*pblocks* specifies the number of 16K blocks of standard memory (i.e., in the first 640K of RAM) to be reserved as the window through which portions of expanded memory appear to MS-DOS.

☐ **Note** If you allocate too many blocks in standard memory, there will not be enough memory to run your application programs.

/E:*eblocks* specifies the number of 16K blocks of extended memory to be devoted to VMEM. This amount of memory should be as large as possible, but must at least equal the number of *pblocks* reserved in standard memory.

This configuration command:

```
DEVICE = VMEM.SYS
```

activates expanded memory emulation according to the VMEM's default values: four physical blocks (totaling 64K) in standard memory are devoted to the "window" and all available extended memory is devoted to emulating expanded memory .

The following configuration command includes all three of the VMEM.SYS parameters:

```
DEVICE = VMEM.SYS /L:128 /P:8 /E:24
```

The command has these effects: VMEM will not use the first 128K in extended memory (/L:128); in standard memory 128K (eight 16K blocks) is devoted to the expanded-memory window

(/P:8); and in extended memory VMEM will use 24 blocks of 16K each, totaling 384K (/E:24).

Disabling and Reenabling the VMEM Memory

In case one of your application programs interacts badly with VMEM, you can temporarily disable the VMEM memory. At the MS-DOS system prompt, enter this command

```
C>VMEM OFF
```

before running the incompatible application program. After you exit the application program, you can reenable the VMEM memory with this command:

```
C>VMEM ON
```

☐ **Note** These memory disabling and reenabling functions aren't performed by the VMEM.SYS device driver; when you enter these two commands, the VMEM.COM command file must be available on disk.

Combining CACHE, VMEM, and VDISK in Extended Memory

The CACHE command and the VMEM.SYS and VDISK.SYS device drivers are designed to not interfere with each other in extended memory. All three can be used simultaneously in any combination without one program overwriting the memory used by the others.

The only guideline you must follow is that you must specify a memory size for VMEM.SYS if you also want to create a virtual disk in extended memory with VDISK.SYS.

For example, if your system had available 384K of extended memory, your CONFIG.SYS file could contain these two commands:

```
DEVICE = VDISK.SYS 200 128 64 /E
DEVICE = VMEM.SYS /E:4
```

(It doesn't matter which of these commands is written first in the CONFIG.SYS file.) The command for VMEM sets aside four 16K blocks of extended memory (totaling 64K) for EMS emulation, which leaves plenty of room for the 200K virtual disk created by VDISK.SYS.

After the DEVICE commands in the above example are executed, you could still enter this CACHE command:

```
C>CACHE ON /E /S:120
```

which creates a cache with 120K of extended memory (all that remains from the initial 384K).

As you can note in this example, to enable CACHE, VMEM, and VDISK to coexist in extended memory, you only have to specify the amounts of memory each uses—you don't have to control the starting location of each command's portion of memory.

However, software from other vendors might deal with your system's extended memory in a way that interferes with CACHE, VMEM, and VDISK. In that case, you might be able to resolve conflicting demands for memory by writing your CACHE and VMEM commands with an /L parameter, which can specify a safe starting location for the portion of extended memory devoted to each program.

Improving the Efficiency of Hard-disk Storage

When a newly formatted disk is empty, MS-DOS copies files to the disk in contiguous blocks; all parts of a file are in adjacent portions of the disk. As you use the disk more, however, the files you delete free up scattered parts of the disk that MS-DOS starts to use again for storage. To do so, MS-DOS breaks up new or revised files into portions that it stores wherever it can find room on the disk, without regard for keeping all parts of a file close together on the disk.

MS-DOS is able to access these fragmented files, but disk operations may slow down if the drive's read/write head has to move to more places on the disk to locate each portion of a file.

Caution Before running DISKPACK, observe these cautions:

- If your hard disk contains copy-protected software, don't use the DISKPACK command without first consulting your application program's manual. After DISKPACK moves files, copy-protected software may no longer work, and you may be unable to reinstall it. The application program may require you to "uninstall" the software before running DISKPACK and reinstall it afterwards.
- Only use this command if you have followed backup procedures for your hard disk and have spared-out bad tracks when you prepared the hard disk.

Do not run DISKPACK while other programs can access the disk (e.g., in a networking or multiuser environment). Do not use DISKPACK to reorganize partitions of operating systems other than MS-DOS.

- DISKPACK must be run by itself, as shown in this guide's examples; never run it in combination with application programs and other MS-DOS commands. In particular, do not run DISKPACK:
  - If DRIVER.SYS has been loaded by the CONFIG.SYS file
  - When any ASSIGN, JOIN, or SUBST commands are in effect
  - If the SHARE command has been executed
  - While any files remain in the PRINT command's printing queue
  - If you have loaded any terminate-and-stay-resident (TSR) programs (a TSR program is one that can be called up or continue to operate while another program runs)

  You can remove any doubt about the status of these commands and programs by running DISKPACK only after rebooting the system from your original Enhanced MS-DOS diskette (i.e., a system diskette that does not contain a CONFIG.SYS or AUTOEXEC.BAT file).
- Before you run DISKPACK, see the instructions on how to recover if a power failure occurs during disk reorganization.

Displaying Statistics About a Disk

To display statistics describing the condition of a disk, but without rearranging any files, enter the DISKPACK command followed by the drive designator and the /S parameter, which calls for the program to produce a detailed, directory-by-directory report. For example, to display a report about drive C, enter this command:

```
C>DISKPACK C: /S
```

If the report indicates disk fragmentation greater than five percent, reorganizing the hard disk with DISKPACK will cause a noticable improvement in the computer's performance.

Reorganizing the Disk With DISKPACK

To reorganize the files on the hard disk, enter the DISKPACK command followed by the drive designator of the hard disk. For example, this command:

```
C>DISKPACK C:
```

tells the operating system to reorganize drive C. Before it starts reorganizing, you'll be prompted to put a blank, formatted diskette in drive A. This diskette is used for temporary storage of data during the reorganization.

While DISKPACK is running, you'll see messages indicating the progress of the file reorganization. DISKPACK may require an hour or more to reorganize a hard disk, depending on how much data you've stored. When this process is finished, you'll see a completion message and the system prompt will reappear.

Recovering from a Power Failure during DISKPACK

If your computer loses power while DISKPACK is reorganizing its hard disk, you must follow this procedure *before you do anything else to the hard disk.* Enter this command:

```
A>DISKPACK C: /R
```

Put the original temporary storage diskette (the same diskette that was used when the interruption occurred) in drive A when DISKPACK asks you to do so. Then the DISKPACK process will resume where it left off.

Controlling the Processor Speed

Some computer systems have a dual-speed processor. If yours does, you can enter the SPEED command to control this feature.

Standard processing speed is the IBM-compatible speed. Using the higher processing speed, however, greatly reduces the time it takes the computer to perform operations.

The SPEED command switches between the standard and high processing speeds; it also can enable or disable the front-panel speed button (if your system has one).

Changing the Processor Speed

When you turn on your computer system, the speed initially defaults to the speed that you selected when you ran the SETUP program. To select high-speed processing, enter the SPEED command like this:

```
A>SPEED H
```

To return to the standard speed, enter this command:

```
A>SPEED S
```

Enabling and Disabling the Front-Panel Button

The front-panel button that controls processor speed is enabled until you disable it with the SPEED command. You may want to disable the pushbutton to prevent anyone from changing the speed during processing; later, after processing ends, you can reenable the button.

The command **SPEED E** enables the front-panel button; **SPEED D** disables it. You can include S or H with E or D on the same SPEED command line. For example, the following command selects high-speed processing and disables the front-panel button:

```
A>SPEED H D
```

You can enter the SPEED command options in any order. The SPEED command changes the processor speed whether or not the front-panel button is enabled.

Controlling the Liquid Crystal Display (LCD)

Your computer system may have an optional 16-character liquid crystal display (LCD) panel for displaying date or time information, the system performance, the current disk drive, and user-generated messages.

This section explains how to use the LCDPANEL command to display the date, the time, the performance graph, and personal messages. When you enter an LCDPANEL command to display the date or time, you can select the display format.

Each of these command options turns on the LCD if it is off. The **LCDPANEL OFF** command turns off the LCD if it is on.

The LCDPANEL command is an external command, so the file named LCDPANEL.EXE must be accessible when you enter this command.

☐ **Note** You do *not* need to install the LCDPANEL.SYS device driver before entering any LCDPANEL commands. All LCDPANEL commands are executed by the LCDPANEL.EXE command file, not the device driver.

Displaying the Date or Time

To display the date on the LCD panel, enter the LCDPANEL command and include its DATE option. The operating system uses the day-month-year format for the date in LCD displays (for example, 29Apr87).

To display the current date, enter the command:

```
A>LCDPANEL DATE
```

The date, processing speed (in megahertz), and the designator letter of the disk drive currently in use appear on the LCD; for example:

```
29Apr87  8MHzC
```

To display the time on the LCD panel, enter the LCD command with the TIME option:

```
A>LCDPANEL TIME
```

The operating system defaults to a 12-hour format, which you can change to a 24-hour format with this command:

```
A>LCDPANEL TIME /24
```

To change back to the 12-hour format, include /12 instead.

You can also use the front-panel controls to toggle between the time and date displays. If neither the time nor the date is being displayed on the LCD, pressing the button displays the time.

Displaying a Performance Graph

To display the performance graph, enter the LCDPANEL command with its GRAPH option, like this:

```
A>LCDPANEL GRAPH
```

The graph may be longer or shorter, depending on your processor speed and the types of operations being performed. When the graph extends to the right, the computer's microprocessor is performing a greater number of operations (e.g., computations) in the measured time. When the graph contracts to the left, less of the computer's processing capacity is being used (e.g., the microprocessor performs fewer computations while it waits for input and output operations).

You can also use the front-panel controls to switch between displaying the performance graph and other information.

Displaying Personal Messages

You can enter the LCDPANEL command with the ECHO option to display a personal message on the LCD. This message can contain both upper- and lower-case letters. The command line includes the message you want to display; for example:

```
A>LCDPANEL ECHO Appointment 1:00
```

If the message is 16 characters or fewer (as in this example) it appears on the LCD and remains until you replace it with other information. If the message is 17 characters or more, it scrolls across the LCD; after the last character of the message scrolls off the LCD, the time, date, or graph information that was displayed previously is restored.

Displaying Output with LCDDEV

With the LCDPANEL ECHO command, what you specify to appear on the LCD panel can only be a literal message, spelled out on the command line. However, if you include the LCDPANEL.SYS device driver in your CONFIG.SYS file, you can use the LCDDEV device name to display any output generated by an MS-DOS command on your LCD panel. For example, the contents of your AUTOEXEC.BAT file will scroll across the LCD panel if you enter this command:

```
A>COPY AUTOEXEC.BAT LCDDEV
```

Or you can redirect a directory listing from the console to the LCD panel with this command:

```
A>DIR B:*.COM > LCDDEV
```

Turning Off the LCD

Turning off the LCD increases system efficiency slightly; in fact, certain applications with high rates of data transfer may require that the LCD be off. To turn off the LCD, enter the command:

```
A>LCDPANEL OFF
```

To turn on the display again, enter one of the LCD commands already discussed.

Activating the System's Combination Lock

Whether or not your computer hardware has a key-operated lock, your system may allow you to lock it through software. It lets you lock the computer by entering a four-digit combination. Until you enter your combination again to unlock the computer, the system is protected from unauthorized use.

While your computer is locked, it can continue ongoing processing and network operations. But the computer's locked state and your combination remain active even if the computer is rebooted or turned off and then on again. The lock will continue to prevent keyboard access until you enter the correct combination.

The LOCK command enables you to select a four-digit combination from any of 10,000 possibilities, to lock the system through software and unlock it by entering your combination, and to change the existing combination.

Caution If you have locked your computer and then forget your combination, the LOCK command and MS-DOS provide no way to unlock the computer. *Do not lose your four-digit combination!* If you cannot unlock your computer, contact your dealer.

Setting a Combination

To set the combination of your lock for the first time, enter the LOCK command with the /S option. For example:

```
A>LOCK /S
```

If no combination has been previously set, the system prompts you to enter one. The prompt shows four shaded squares, one for each digit of the combination. The cursor is in the first square.

If you don't want to set a combination, press Enter without typing anything. MS-DOS returns to the system prompt.

When you select a combination, you may want to enter one that you can conveniently remember—and that only you know—such as your personal identification number for a bank's automatic teller machine. To enter the combination, type four digits and press Enter; for example:

```
Enter new combination: 7 4 3 9
```

☐ **Note** The numbers you type do not appear on your screen. Be sure you write down your combination and store it in a safe place.

After you enter the combination, the operating system prompts you to verify this combination by reentering it.

Reenter the combination. As before, the operating system highlights the four shaded squares, but displays no numbers. It then returns to the system prompt.

☐ **Note** If you try to lock the keyboard when a combination has not been previously set, an error message appears telling you to run **LOCK /S** to set a combination first.

Activating the Lock

After you set a combination, you can lock the keyboard by entering the LOCK command without any options or by pressing the LCD front-panel pushbuttons. The operating system prevents unauthorized locking by prompting you to enter a valid combination. Once locked, the computer can't accept commands from the keyboard. You may then leave the computer on or turn it off, as you wish.

If your computer has an LCD with front-panel controls, you can begin the lock procedure by pressing the three control buttons simultaneously. The advantage of this approach over entering the LOCK command is that you can secure your system while an application program is running without having to return to the system prompt.

You can also begin the lock procedure by entering this command:

```
A>LOCK
```

Whichever way you began the lock procedure, the operating system next prompts you to enter the four-digit combination.

At this point, if you want to end the procedure without locking the keyboard, you can press Enter. The system prompt reappears.

To lock the keyboard, type the four digits that you assigned when you set the combination, and then press Enter. For example:

```
LOCK: 7 4 3 9
```

If you enter an incorrect combination, an error message prompts you to try again. To abort the operation after the first or second try, you need to reboot the operating system—pressing Enter no longer works after you enter the first combination. If you fail to enter a valid combination after three tries, the operating system tells you that you must reset (i.e., reboot) the operating system to return to normal MS-DOS operation.

When you enter a valid combination, the operating system displays this message in the upper right corner of the screen:

```
SYSTEM LOCKED
```

You may now turn off the computer system. When you turn the computer system back on, the SYSTEM LOCKED message reappears.

☐ **Note** When the computer system is locked with power on, the operating system prompt appears in its normal position on the left side of the screen. (This may not be the case if you were running an application program at the time of locking.) Although the keyboard is locked, the system unit can still function for network operations or other nonkeyboard input.

Unlocking the System

When the screen displays the SYSTEM LOCKED message, the keyboard doesn't respond to input. You can begin the sequence that unlocks the keyboard either by pressing Esc or, if your computer system has the optional LCD panel, by pressing the three front-panel controls simultaneously.

The operating system responds by displaying this message in the upper right corner of the screen:

```
UNLOCK:
```

If you don't remember the combination, you can abort the procedure now by pressing Enter. If you want to proceed, type the combination and then press Enter; for example:

```
UNLOCK: 7 4 3 9
```

The system doesn't display the numbers you type. If the combination is not correct, the operating system prompts you to reenter it. In that case, you have two more tries. If the combination is still not correct, the operating system prompts you to reboot MS-DOS and try again.

If the combination you enter is correct, you see this message:

```
SYSTEM UNLOCKED
```

You may now proceed with normal processing.

Changing the Combination

If you know the current combination, you can change it by entering the LOCK command with the /S option, just as you did when setting the combination initially; for example:

```
A>LOCK /S
```

Because the system already has a combination, it verifies that you have the authority to change the old combination by prompting you to enter it.

You can abandon the operation before entering a combination by pressing Enter. If you continue and enter the old combination successfully, MS-DOS prompts:

```
Enter new combination:
```

Enter the new combination, as already described. For confirmation, the system asks you to reenter the new combination. When you enter the correct four digits, MS-DOS returns to the system prompt.

Appendix
A Command Descriptions

This appendix contains brief descriptions of *all* MS-DOS commands. It is divided into three sections: standard MS-DOS commands, batch-processing commands, and CONFIG.SYS commands.

The online HELP feature provides screens with more detailed information about each of the commands listed here. To see this information, enter HELP followed by the name of the command.

Standard MS-DOS Commands

The following commands can be entered at the MS-DOS system prompt. The symbol (I) after the command name specifies that the command is internal to MS-DOS; (E) specifies an external command that is contained in a separate program file (i.e., a file with a .COM or .EXE file name extension).

APPEND (E)	Specifies directory paths for MS-DOS to follow when searching for a data file. In a network, this command lets you locate files on remote storage devices.
ASSIGN (E)	Reassigns a disk drive designator to a different disk drive; for example, if you assign the drive designator A to drive C (the hard disk), the operating system accesses drive C whenever a command refers to drive A.
ATTRIB (E)	Establishes, displays, and removes write-protection and archiving attributes of files.
BACKUP (E)	Makes a backup copy of a hard disk directory onto a diskette; used in combination with RESTORE.
BREAK (I)	Determines the conditions during which MS-DOS will check if the user has given a break signal by pressing Ctrl Break or Ctrl C.

CACHE (E)	Reduces the frequency of disk access by storing in RAM recently read or written disk contents.
CHDIR (I)	Changes from the current directory to the directory you identify; if you specify no new directory, the CHDIR command displays the current directory path name. (CD is a synonym for the CHDIR command.)
CHKDSK (E)	Analyzes the specified disk drive, and displays a report on that disk and the system's memory.
CLS (I)	Clears the screen.
COMMAND (I)	Loads the command processor program COMMAND.COM (which runs MS-DOS internal commands) from the current directory or a specified directory; also reloads the transient portion of the program if an application program overwrites it during processing. Can specify a larger amount of memory to be used by the operating system's environment.
COPY (I)	Copies one or more files to another location on the same disk or on another disk. You can also combine several source files into one target file.
CTTY (I)	Changes the terminal device from which you issue MS-DOS commands, such as from the console (CON) to a device attached to the auxiliary I/O port (AUX).
DATE (I)	Displays the system date and enables you to change it. MS-DOS records the date in the file directory when you create or change a file.
DEBUG (E)	Enables you to test and correct program files.
DEL (I)	Deletes one or more specified files; synonymous with the ERASE command.
DIR (I)	Displays a listing of the contents of a disk or directory; also enables you to obtain information about a specific file.

DISKCOMP (E)	Compares the first diskette designated in the command (the source disk) with the second diskette designated (the target disk).
DISKCOPY (E)	Creates a target diskette that is an exact duplicate of a source diskette. If the target diskette is unformatted or not the same format as the source disk, DISKCOPY will automatically format the target disk before copying data from the source. This command cannot by used with hard disks and cannot be run over networks.
DISKPACK (E)	Recopies fragmented disk files and rearranges them into contiguous blocks and sequential order to improve file access.
EDLIN (E)	Enables you to create or edit text files; contains its own set of commands with which you can add, insert, list, move, or erase lines from the file.
EQUIP (E)	Displays a list of installed equipment and indicates the number and types of memory, processor, monitor, disk drive, and I/O components.
EXE2BIN (E)	Converts executable files (e.g., files with an .EXE file extension or the equivalent) to files in binary format.
EXIT (I)	Leaves the MS-DOS command level (COMMAND.COM) to return to a previous operating environment, such as an application program, if one exists. This command assumes you temporarily left the application to perform an MS-DOS function, and are exiting MS-DOS to resume the application.
FC (E)	Compares two files and displays the results on the screen or stores them in a result file. Command options let you disregard spacing and capitalization in ASCII comparisons.
FDISK (E)	Enables you to create, change, display, or delete MS-DOS partitions on the hard disk before formatting.
FIND (E)	Searches for and reports on the occurrence of a character string in one or more files. Command options let you display the names of files *not* containing the string, a count of the lines containing

the string in each file, and the relative line numbers of the lines containing the string.

FORMAT (E)	Sets up the structure of the disk (hard disk or diskette) so it can accept MS-DOS files. Also enables you to copy MS-DOS system files onto the disk and to assign a volume label (or disk name) to the disk at the time it is formatted. This command cannot be run over networks.
GRAFTABL (E)	Loads into memory a table of ASCII graphics characters 128 through 255, which can then be displayed in the 640 x 200 graphics mode.
GRAPHICS (E)	Enables you to print copies of graphic screen images that are displayed on a color monitor or on a monochrome monitor with a graphics adapter.
GWBASIC (E)	Starts the GW-BASIC programming language interpreter, and enables you to enter GW-BASIC statements and commands.
HDINIT (E)	Initializes the hard disk before you partition and format it.
HDPARK (E)	Moves the read/write head of the hard disk (or disks) to a track where no data is stored, thus protecting the head when you move the disk drive unit. When you subsequently turn on the system, the read/write head automatically "unparks" and the hard disk is ready for use.
HELP (E)	Lists the available MS-DOS commands. If you follow the HELP command with the name of another command, it provides online reference information about that command.
JOIN (E)	Temporarily connects an empty subdirectory of the root directory to a second disk drive, making files on that drive accessible through the subdirectory. Not available over networks.
KEYB*xx* (E)	Establishes an international keyboard configuration, where the characters *xx* stand for a two-character country code. Available configurations include French (KEYBFR), German (KEYBGR), Italian (KEYBIT), Spanish (KEYBSP), and United Kingdom (KEYBUK).

LABEL (E)	Adds, changes, displays, or deletes the volume label of a disk in the specified disk drive. Not available over networks.
LCDPANEL (E)	On systems with a liquid crystal front-panel display, turns on or off the display, specifies the time in a 12-hour or a 24-hour format, and displays system performance graphs or personal messages.
LINK (E)	Lets you link program object modules to produce machine-language code required for the computer system to run programs.
LOCK (E)	Prevents unauthorized access to the computer system by disabling keyboard input, except for the four-digit combination that "unlocks" the keyboard again.
MKDIR (I)	Creates a new directory. (MD is a synonym for the MKDIR command.)
MODE (E)	Defines the operational characteristics of input/output devices (such as printers and modems), assigns I/O ports, and sets up asynchronous communications parameters.
MORE (E)	A command filter that causes the system to display screen output one screen at a time, pausing at each screenful until you indicate you are ready to see the next screen. For example, the command **DIR ¦ MORE** (with the vertical bar separating the DIR and MORE commands) displays the disk file directory one screen at a time.
PATH (I)	Tells MS-DOS to look in the specified directories for external commands not found in the current directory. This command also displays the current directory path, and can restrict the search for external commands to the current directory.
PRINT (E)	Queues files to a printer, and then prints them in sequence while you run other MS-DOS commands. Command options let you select the printer, set the print buffer size, delay printing, limit the number of files in the queue, cancel previous print requests, and add new files to the queue.

PROMPT (I)	Changes the MS-DOS system prompt.
RECOVER (E)	Recovers a file on a disk with defective sectors, and recovers all files on the disk if the directory contains defective sectors; however, does *not* recover data in the defective sector. MS-DOS reads the file or disk sector by sector, marking defective sectors to prevent data from being recorded there in the future. RECOVER is not available over networks.
RENAME (I)	Changes the name of one or more files. (REN is a synonym.)
REPLACE (E)	Copies updated files in the source directory over current versions of the same files in the target directory. Command options let you include files *not* previously in the target directory, replace files only if the source files are newer than the target files, search all target subdirectories to replace matching files, and select which target files to replace.
RESTORE (E)	Copies files from one or more diskettes created with the BACKUP command onto the target hard disk.
RMDIR (I)	Removes a directory from the directory tree structure. (RD is a synonym for the RMDIR command.)
SELECT (E)	Creates an MS-DOS disk that automatically loads a selected international date, time, and currency format and keyboard configuration when you boot the operating system.
SET (I)	Allows you to examine and alter the contents of the strings in the operating system's environment.
SHARE (E)	Enables network file sharing, establishes the size of the memory area that MS-DOS uses to keep track of file-sharing information, and defines the number of file locks used in file sharing. SHARE applies only to systems with active networking.
SORT (E)	Sorts input (such as a list of names or numbers in a data file) in ascending or descending alphabetical or numerical order, and writes the sorted information in a designated file. As a command filter, sorts the output from a command; for example, the

command **DIR ¦ SORT** rearranges the file directory in alphabetical order. SORT can also display a sorted data file.

SPARE (E)	Tests the hard disk for bad recording tracks and creates a "bad track list" on the hard disk to prevent MS-DOS from writing data to those tracks.
SPEED (E)	For computer systems with a dual-speed system clock, toggles the clock speed between standard speed and high speed and displays the current speed setting. This command can also enable or disable the dual-speed operation.
SUBST (E)	Substitutes a disk drive designator for a complete directory path, enabling you to access the directory path by entering the substituted disk designator. This command also displays the directory substitutions in effect, and cancels substitutions. SUBST is not available over networks.
SYS (E)	Transfers the two hidden system files from the disk in the default drive to the blank, formatted disk in the specified drive or updates existing system files on the disk in the specified drive.
TIME (I)	Displays the system time and enables you to change it. MS-DOS records the time in the file directory when you create or change a file.
TREE (E)	Displays the directory tree structure of a disk by displaying the path of each subdirectory; can also include file names in the directory path listing.
TYPE (I)	Displays the contents of a text file.
VER (I)	Displays the current MS-DOS version number.
VERIFY (I)	Checks that all files copied to disk are intact (for example, that there are no bad sectors), and displays an error message if problems exist. This command toggles the setting between **VERIFY OFF** (the default setting) and **VERIFY ON**. If you enter **VERIFY** by itself, it displays the current setting.

VMEM (E)	Hides or reveals the portions of memory devoted to the virtual expanded memory set up with the VMEM.SYS device driver. The command **VMEM OFF** can hide the virtual expanded memory from an incompatible program; **VMEM ON** reveals it again.
VOL (I)	Displays the volume label of the disk in the specified drive; if no label exists, displays a message to that effect.
XCOPY (E)	Copies files, directories, and subdirectories containing files from the source drive to the specified target drive.
Batch-Processing Commands	The following commands are designed to be included in batch files to control the execution of other commands. All these batch-processing commands are internal commands.
ECHO	The command **ECHO ON** or **ECHO OFF** causes subsequent command lines in a batch file to be printed (or not) on the screen as they are executed. Can also print to the screen a message you write after the word ECHO on the command line.
FOR	Controls conditional execution of command lines in a batch file.
GOTO	Controls execution of commands within a batch file.
IF	Allows conditional execution of commands in a batch file.
PAUSE	Causes the operating system to pause before executing further commands until the user presses a key.
REM	Allows you to include remarks in a batch file. The text that follows REM on the command line will be displayed when the batch file is executed if ECHO ON is in effect.
SHIFT	Allows a batch file program to deal with more than ten replaceable parameters (which are normally identified as %0 to %9).

CONFIG.SYS Commands	These configuration commands can be executed only if you enter them in a file named CONFIG.SYS, which is executed at the time the computer is booted up. All configuration commands are internal commands.
BREAK	Sets the break-checking status when the computer is booted up.
BUFFERS	Sets the number of temporary memory areas where MS-DOS holds data being read from or written to disk.
COUNTRY	Causes the operating system to use time, date, and currency formats for the specified country.
DEVICE	Installs the specified device driver, which is an external file. The following device drivers are provided with Enhanced MS-DOS. **ANSI.SYS** Enables the computer to process ANSI escape sequences. **DRIVER.SYS** Installs an external disk or tape drive. **LCDPANEL.SYS** Establishes LCDDEV as the device name of the 16-character LCD front panel. **VDISK.SYS** Creates a virtual disk drive in RAM. **VMEM.SYS** Causes extended memory to emulate EMS and EEMS compatibility.
DRIVPARM	Defines alternative settings for drive devices.
FCBS	Sets the maximum number of files that file control blocks (FCBs) can have open at the same time.
FILES	Sets the maximum number of files that MS-DOS can have open at the same time.
LASTDRIVE	Sets the last drive designator letter that MS-DOS accepts as valid.
SHELL	Loads an alternative command processor in place of the default COMMAND.COM.

Appendix

B Batch Files

In Chapters 1 through 7 of this guide, you used MS-DOS commands interactively; that is, you entered a command, and MS-DOS responded by immediately executing it. You will do most of your work with MS-DOS in this *interactive* mode.

Another way to process commands is in *batch* mode. In this mode, you collect commands into a special type of file called a batch file. When you tell MS-DOS to access the batch file, the operating system reads each line in it and then executes the commands, one after the other.

Batch files can be very simple, perhaps one or two commands. They can also be very elaborate, executing some commands and bypassing others, depending on the contingencies specified in the file. Batch files provide you with a simple way to perform routine, repetitive, complex, or inconvenient operations by entering a single command.

This appendix shows you how to create a simple batch file and run it automatically every time you boot MS-DOS.

Creating a Batch File

A batch file contains a sequence of commands and their parameters. The file can have any name you want to give it, but it must always have the extension .BAT (for "batch").

Now, suppose you have a procedure that you always perform when you boot MS-DOS at the start of the day, such as changing from the root directory to a subdirectory and running a word-processing application program. Rather than enter the commands yourself, you can include them in a batch file that automatically runs whenever you start the operating system.

For a batch file to run automatically when you boot up the computer, it must have the name AUTOEXEC.BAT and be in the root directory of the disk from which you boot MS-DOS. The file can contain any MS-DOS command and the file names of executable files of applications programs. If you have an AUTOEXEC.BAT file, the usual MS-DOS date and time prompts do not appear unless your file includes lines with the DATE and TIME commands.

The example in this appendix creates an AUTOEXEC.BAT file that turns off the display of commands being executed, sets the operating system prompt to display the current directory, changes to the WP subdirectory, and starts a word-processing program.

To begin, enter the COPY CON command followed by the name of the file; for example:

```
A>COPY CON AUTOEXEC.BAT
```

As in all batch files, you enter one command per line. (See "Creating an AUTOEXEC.BAT File" in Chapter 6 for a more detailed explanation of the COPY CON procedure.)

Controlling the Display of Messages

When you run a batch file, MS-DOS acts the way it would if you entered the commands interactively; that is, it displays the command input, executes the command, and displays command output. The display of input, including the system prompt, is called the *echo* from the computer, because the operating system repeats on the screen what you type at the keyboard.

Echo can be either on or off. When you boot MS-DOS, echo is on by default. To control the echo during the execution of a batch file, you can insert **ECHO OFF** and **ECHO ON** commands in the batch file.

If you don't want batch file input displayed, **ECHO OFF** should be one of the first commands in your file. You can include the **ECHO ON** command in the batch file whenever you do want the lines in the batch file to appear on the screen. To resume normal interactive processing, the operating system turns screen echo back on when batch processing ends.

Including Descriptive Remarks in a Batch File

To help other users understand your batch file, you can include comments or remarks in it. The REM command tells MS-DOS that the line of text that follows is a remark. (If you have more than one line of text, include a REM command for each line.) For example:

```
REM This file runs word-processing software.
```

If a REM command occurs in the batch file when echo is on, MS-DOS displays the command and the remark on the screen.

If you want to include blank lines in the batch file, use REM commands with no text following.

Entering Other Command Lines in a Batch File

The remaining commands in this sample batch file include the PROMPT command (to display the current directory and the > character), the CD command (to change from the root directory to the WP directory), and WPROC, which, for purpose of this example, is the hypothetical name of the executable file that starts up the word-processing application.

```
PROMPT $P$G
CD WP
WPROC
```

To end the file, press Enter to bring the cursor to a blank line, press Ctrl Z to enter an end-of-file marker, then press Enter. You'll see a message that indicates the lines you typed are copied into a disk file named AUTOEXEC.BAT. MS-DOS returns to the operating system prompt.

Displaying the Contents of a Batch File

To display the contents of a batch file, use the TYPE command. For example, to display the batch file you just created, enter:

```
A>TYPE AUTOEXEC.BAT
```

The system displays the complete file contents:

```
ECHO OFF
REM This file runs word processing software.
PROMPT $P$G
CD WP
WPROC
```

Note that if you use COPY CON to revise this file, you have to reenter the whole file. The COPY command replaces the original file with the revised file.

Running a Batch File

To run a batch file, you need only enter the name of the file; you can omit the .BAT file extension. MS-DOS treats the file name as a command to execute the file contents.

You can run the AUTOEXEC.BAT file in this way by entering the name AUTOEXEC as if it were a command. However, the reason you give a batch file the name AUTOEXEC.BAT is so it can run automatically when you boot up the system. If you were to boot up the computer with a disk containing this AUTOEXEC.BAT file (and if the word-processing software were actually in the WP directory), the line in the batch file with the WPROC command would start up that application.

If you need to stop batch file processing before it finishes, press Ctrl Break. MS-DOS asks if you want to terminate the batch job; answer *yes*.

Batch-Processing Commands

There are several commands that are effective only when you include them in a batch file, rather than enter them at the system prompt. In addition to ECHO and REM, which were discussed in this chapter, MS-DOS provides these batch-processing commands: FOR, GOTO, IF, PAUSE, and SHIFT.

To learn how to use batch-processing commands, refer to the online HELP screens about each command.

Appendix

C International Configurations

Your computer's operating system is normally configured for use in the United States. However, MS-DOS allows you to change the computer's national configuration. For example, your system can handle the date and time in the format appropriate to the country you specify. Your keyboard will be able to input letters with accents and diacritical marks required by such languages as French, German, or Spanish.

This appendix provides a table of available national configurations. It then describes how to use the SELECT command to create a bootable diskette that automatically sets up your chosen national configuration whenever you use that disk to boot up the computer. If you have a hard disk, however, you should turn to the final part of this appendix, which explains how to format the hard disk with the SELECT command and also how to set up a national configuration without having SELECT reformat the hard disk.

National Configurations for System and Keyboard

Whichever method you use to configure your system, you will need to refer to a table that lists the available national configurations.

Each two-character keyboard code (except US) refers to an external program with the file name KEYB*xx*.COM, where *xx* is the two-character code from the table in this appendix (for example, KEYBFR.COM is the configuration program for French keyboards). However, no command file is required to install the standard United States keyboard; with the SELECT command you can specify US even though no file named "KEYBUS.COM" exists.

Country	Country Code	Keyboard Code	Keyboard Program
Australia	061		
Belgium	032		
Canadian-French	002		
Denmark	045		
Finland	358		
France	033	FR	KEYBFR.COM
Germany	049	GR	KEYBGR.COM
Italy	039	IT	KEYBIT.COM
Israel	972		
Middle East	785		
Netherlands	031		
Norway	047		
Portugal	351		
Spain	034	SP	KEYBSP.COM
Sweden	046		
Switzerland	041		
United Kingdom	044	UK	KEYBUK.COM
United States	001	US	(none required)

Configuring a System Diskette with Select

The procedures in this section assume that you have a dual-diskette system with drive A as the current drive. If you have only one diskette drive, prompts will appear telling you when to insert the source diskette (intended to be in drive A) and the target diskette (intended to be in drive B).

1 Insert your MS-DOS system diskette in drive A and a blank diskette in drive B. Now enter the SELECT command, followed by a country code (three digits) and a keyboard code (two letters). You must include both of these codes with the SELECT command.

For example, to create an MS-DOS diskette that will install the French system and keyboard configuration, enter this command:

```
A>SELECT 033 FR
```

If you want to install a national system (such as for Portugal), but will be using a standard United States keyboard, enter **US** as the keyboard code. For example:

```
A>SELECT 351 US
```

2 The first task performed by the SELECT command is to format your target diskette. SELECT will display a warning message. Reply that you *do* want to continue

3 When SELECT finishes formatting the target diskette and copying the system files onto it, a message will ask if you want to format another diskette. You must answer *no*.

SELECT then copies all the files from the source diskette and creates CONFIG.SYS and AUTOEXEC.BAT files on the target diskette. When SELECT is finished, the system prompt reappears.

Activating the New National Configuration

To have the national configuration of your system and keyboard take effect, reboot MS-DOS from the new diskette in drive A.

Configuring a Hard Disk for International Use

Although the SELECT command can set up the national configuration of a hard disk, keep in mind that SELECT always reformats the target disk. Running that command will destroy all files and information that are already on the hard disk.

If your hard disk has already been formatted and contains many files you want to keep, it may be faster and easier for you to create the appropriate CONFIG.SYS and AUTOEXEC.BAT files directly on your hard disk, without running SELECT.

One way you can create these files is to follow the instructions in Chapter 6 (for CONFIG.SYS) and Appendix B (for AUTOEXEC.BAT). The CONFIG.SYS file should contain a line with the desired country code, such as **COUNTRY=002**. The AUTOEXEC.BAT file should contain the keyboard installation command name, such as **KEYBFR**. (For a standard United States keyboard you do not need to enter a keyboard-installation command.)

The other way is to run SELECT to create a diskette with the desired national configuration, and then copy its CONFIG.SYS and AUTOEXEC.BAT files onto the hard disk.

Configuring a Hard Disk with SELECT

If you decide to run the SELECT command to format your hard disk, follow these directions:

1 You should have already prepared your hard disk through the step of partitioning it with FDISK, but have not yet formatted the hard disk. If you haven't reached this point, follow the instructions for preparing a hard disk given in your computer's installation and assembly manual and in Chapter 1 of this user's guide. When you reach the step of formatting the hard disk, follow the instructions in this section of the appendix instead.

2 Enter the SELECT command, including drive designators that identify the diskette drive (A: or B:) that will hold the system diskette (the source drive) and the hard disk (the target drive). For example:

```
A>SELECT A: C: 033 FR
```

The last two codes are the country code (three digits) and keyboard code (two letters), selected from the table in this appendix. You must include both codes with SELECT.

3 Before it formats your hard disk, SELECT will display a warning message. Respond that you *do* want to continue.

SELECT will then copy all the files from the source diskette onto the hard disk's root directory and will create CONFIG.SYS and AUTOEXEC.BAT files. When SELECT is finished, the system prompt reappears.

4 Remove the system diskette from drive A. In its place insert the MS-DOS supplemental programs diskette. Copy that diskette's files onto the hard disk with this command:

```
A>COPY A:*.* C:
```

5 Remove any diskette from drive A. Reboot MS-DOS from the hard disk to have the national configuration take effect.

☐ **Note** Remember that SELECT creates two files on the target disk: AUTOEXEC.BAT and CONFIG.SYS. If you ever need to modify these files to change the startup configuration of your computer, be sure to add any new commands to the existing versions of these two files. If you were to create a new file with one of these file names, it would overwrite the existing file and remove the national configuration you established with SELECT.

Appendix D

Messages

This appendix explains some of the MS-DOS messages you may encounter, where the meaning of the message may not be clear or where you may not understand what to do in response. This is not an exhaustive list of messages.

Some messages may appear when the operating system is trying to access a device such as a disk drive or printer. If such an error occurs, MS-DOS displays an error message and prompts:

```
Abort, Retry, Ignore?
```

To respond, select one of the following options by pressing the key named:

Option	Description
A	Aborts the operation. MS-DOS returns to the system prompt.
R	Retries the operation. Before you select this option, try to correct the problem that caused the error message.
I	Ignores the error condition and continues the operation. This option can cause loss of data, so use it carefully.

The messages in the rest of this appendix are arranged in alphabetical order. The command or system names in brackets below the message text indicate the source(s) of the message.

`Bad or missing Command Interpreter`
[MS-DOS]

Possible cause MS-DOS cannot find the COMMAND.COM file on the disk. The file may be missing from the root directory, or there may be a problem with the file.

Action Restart the system with a backup MS-DOS diskette. Copy the COMMAND.COM file from the backup diskette to the root directory of the disk that failed.

`Content of destination lost before copy` [COPY]	**Possible cause** This message appears when you enter the COPY command incorrectly and the source file is overwritten before the system can copy it. For example, entering a command like: **COPY FILE1 + FILE2 FILE2** destroys FILE2 before the system can copy it. **Action** The source file is no longer in the directory and is probably lost. If you have previously backed up the lost file, copy the backup file and reenter the COPY command to combine the same files, but with a different file name for the destination file (i.e., the target file).
`Copy not completed` [DISKCOPY]	**Possible cause** This message means that DISKCOPY cannot copy the entire disk. Either the source or target diskette has physical errors. **Action** Repeat the command with a different target disk or a backup of the source disk.
`Diskette/Drive not compatible` [DISKCOPY]	**Possible cause** The source and the target drives aren't compatible; for example, you tried to use DISKCOPY to copy from a diskette to the hard disk or to a diskette of another size. **Action** Use a target diskette that is the same type and size as the source diskette.
`Disks must be the same size` [DISKCOPY]	**Possible cause** With the DISKCOPY command, you can't copy the contents of a diskette to a disk with a different data storage format; for example, you can't copy a 5.25-inch diskette to a 3.5-inch diskette or to the hard disk. **Action** Use the COPY or XCOPY command to copy files onto dissimilar disks.
`Error. Hard disk needs to be initialized` [FDISK]	**Possible cause** You tried to execute the FDISK command without first using the HDINIT command to initialize the hard disk. **Action** Be sure you prepared your hard disk as described in the installation and assembly manual for your computer system. Then reissue the FDISK command.

Fatal error with hard disk Hard disk is not initialized [FORMAT]	**Possible cause** You tried to format the hard disk without first preparing it for formatting. A "fatal error" means the command can't continue. The FORMAT command ends and returns to the system prompt. **Action** First, use FDISK to set up the operating system partition before you reissue the FORMAT command. If that doesn't resolve the problem, follow the disk preparation instructions in the installation and assembly manual for your computer system, and then try again.
File creation error [RESTORE]	**Possible cause** This message appears if you try to add a new file to a full directory in the target disk. **Action** Execute the CHKDSK command to ascertain the condition of the disk directory. No action is necessary for the attempt to replace read-only files.
Incorrect DOS version [BACKUP] [FORMAT] [RESTORE]	**Possible cause** Many of the commands for this version of MS-DOS don't work with older versions of MS-DOS. The commands here (and some others) will run only under the exact version of MS-DOS for which they were configured. **Action** Reboot MS-DOS using the correct version of the system, and reenter the command.
Invalid device [MS-DOS]	**Possible cause** The device name you specified in a command is not valid. **Action** Reissue the command using one of the device names listed in Chapter 2, "The Fundamentals of MS-DOS."
Invalid filename or file not found Invalid drive or filename Invalid drive specification [MS-DOS]	**Possible cause** These messages indicate that you entered an invalid disk drive designator or file name in a command. **Action** Reenter the command and specify a valid drive designator, and file name. If necessary, specify the complete path name for the directory where the file is located. Be sure the disk drive designator has a colon after the drive letter. Check file names for valid characters and name length.
Invalid media or track 0 bad - disk unusable [FORMAT]	**Possible cause** The diskette you're trying to format may not be the correct type for your computer.

	Action Be sure you are using the correct type of diskette for your computer; consult your computer system manual for information about the correct diskettes to use for your computer. Refer to the discussion of the /4 parameter in "Formatting Diskettes" in Chapter 1.
`Invalid parameter` [BACKUP] [CHKDSK] [FORMAT] [RESTORE]	**Possible cause** You specified one or more invalid options or parameters in a command. This message also appears if you omit the colon from a disk drive designator in these commands. **Action** If the program expects a disk drive designator, be sure to enter a colon following the drive letter. In other cases, make sure the character following the slash (/) is valid for the program being run.
`Invalid parameter for hard disk` [FORMAT]	**Possible cause** You entered an incorrect disk drive designator for the hard disk; for example, you may have entered **FORMAT E:** when drive E doesn't exist. **Action** Reenter the command using the designator for an existing disk drive.
`Invalid path,not directory or directory not empty` [MS-DOS]	**Possible cause** With the RMDIR command, you can't remove the specified directory because of one of the following conditions: • One of the names you specified in the path was not a valid directory name. • The directory you specified still contains entries for files or other subdirectories (aside from the . and .. markers). • You tried to remove the current directory. **Action** Correct the invalid directory name in the path, or delete the appropriate files or subdirectories. If you entered the command from the current directory, change directories.
`Invalid working directory` [MS-DOS]	**Possible cause** The disk is defective. **Action** Replace the disk. Make a new working copy of the disk's contents from your backup copy, if you have one available. If you do not have a backup copy and some files on the defective disk cannot be copied successfully in any other way, use the RECOVER program.

Memory freed is not available because of allocated memory after CACHE [CACHE]	**Possible cause** After CACHE allocated buffer space for disk caching, you loaded a terminate-and-stay-resident (TSR) program that resides above CACHE's buffer. When you turn off CACHE, this message appears to tell you that the memory CACHE used still won't be available to other programs. **Action** To recover the memory released by CACHE, you must unload the other TSR program and then reload it. If that program provides no way to unload itself, you may need to reboot without running CACHE to regain use of that portion of memory.
Non-System disk or disk error Press a key to retry [Hardware]	**Possible cause** This message appears if: • You didn't insert a diskette in drive A. • The diskette in drive A doesn't contain a valid copy of the MS-DOS system files. • Something is wrong with the diskette in drive A that is preventing a normal boot. • The hard disk in a hard-disk system doesn't contain the necessary software to boot or has an internal problem that won't allow a boot. **Action** Make sure you have inserted a system diskette in drive A. If you did not, insert one now and press any key to retry. If you want to boot from the hard disk, make sure the necessary files are on the disk. If you think something may be wrong with the system diskette in drive A, try inserting another copy of the system diskette and press any key. If the message persists, call your service representative.
No space for a cylinder partition at cylinder # [FDISK]	**Possible cause** The partition size you entered cannot be created at the starting cylinder you entered. **Action** Either enter a smaller partition size, or start the partition at a cylinder number that will allow for the size you entered. The **HELP FDISK** command provides reference information about FDISK.

Pathname too long [MS-DOS]	**Possible cause** You specified a path name that exceeds 64 characters. **Action** Reenter the path name correctly if you mistyped it, or change directories and reissue the command.
Restore file sequence error [RESTORE]	**Possible cause** A file was not restored because the backup diskettes were not restored in sequential order. **Action** Reissue the RESTORE command and insert the diskettes in sequential order.
Source and target drives are the same [BACKUP] [RESTORE]	**Possible cause** You entered the same disk drive designator for both the source and target drives. For these commands, the source and target drives must be different. **Action** Reissue the command, specify source and target drives that are different. For example, you can specify drive A as the source drive and drive C as the target drive.
Source does not contain backup files [RESTORE]	**Possible cause** The source diskette does not contain files created by the BACKUP command. **Action** Insert the diskettes with the files created by the BACKUP command, then reissue the RESTORE command.
System files restored Target disk may not be bootable [RESTORE]	**Possible cause** If the diskettes you restored contained an earlier version of MS-DOS, the system may not be able to boot with the restored files. **Action** If you can't boot from the disk containing the restored files, use the SYS command to copy the MS-DOS system files to that disk. Also, copy COMMAND.COM to the same disk. You should then be able to boot. For more information, consult the SYS command documentation in the online HELP feature.
Target can not be used for backup [BACKUP]	**Possible cause** BACKUP failed to copy files to the target diskette, and returned to the system prompt. **Action** Reissue the BACKUP command after you insert a different target diskette in the drive.

Target is full [RESTORE]	**Possible cause** The disk to which you are restoring files has no more file storage space. The command aborts and returns to the system prompt. **Action** Delete unneeded files from the target disk and reissue the RESTORE command.
The last file was not restored [RESTORE]	**Possible cause** RESTORE command processing was stopped before the last file was restored, or RESTORE could not restore the last file because the target disk is full. **Action** If you stopped the execution of the RESTORE command, reissue the RESTORE command, supplying only the names of the files or directories that were not restored before. If the target disk is full, delete unnecessary files from the disk and then reenter the RESTORE command, specifying only the names of the files or directories not already restored.
Track 0 bad - disk unusable [FORMAT]	**Possible cause** Track 0 is where the boot record, file allocation table, and directory must reside. If this track is bad, you can't use the diskette. **Action** Refer to the FORMAT command description in the online HELP feature. Check to be sure you're using the correct disk type, or use a new diskette.
Unable to create a directory [MS-DOS]	**Possible cause** MS-DOS cannot create the directory you specified in the MKDIR command. One of the following conditions exists: • The directory you want to create already exists. • One of the directory path names you specified can't be found. • You attempted to add a directory to an already full root directory. **Action** Make sure that a file or directory with the name you specified does not already exist in the parent directory (or current directory). Then, recheck all the directory names to make sure they are valid. Finally, execute the CHKDSK command to see if the directory is full.
Unable to erase *filename* [BACKUP]	**Possible cause** BACKUP could not erase the specified file on the backup disk to add a new file.

	Action Make sure the target diskette is not write-protected; if it is, remove the write protection. Then reissue the BACKUP command.
`Warning! All data in the MS-DOS partition will be DESTROYED. Do you want to continue . . . .? [N]` [FDISK]	**Possible cause** You are attempting to delete an MS-DOS partition with the FDISK command. This message warns you that any files contained in that partition will be erased if you proceed with the deletion. **Action** If the partition to be destroyed contains files that you want to save, be sure to back up the files on diskettes before you continue. In that case, press *N* to abort the FDISK operation. Back up the files, then reissue the FDISK command.
`Warning! Files in the target drive, `*d*`:\ root directory will be erased` [BACKUP]	**Possible cause** The files in the root directory of the backup disk on the specified drive will be destroyed if the backup continues. **Action** If you want to keep root directory files of the target drive, press Ctrl Break to abort the operation. Then back up the source drive onto another backup disk. If you want to erase the root directory files, press any key to continue the backup operation.

`Warning! No files were found to back up`
[BACKUP]
`Warning! No files were found to restore`
[RESTORE]
	Possible cause No files matching the command parameters were found in the specified directory (for BACKUP) or the backup directory (for RESTORE).
	Action Check the command parameters; if they're incorrect, reissue the BACKUP or RESTORE command with the correct parameters.

`Write protect error writing drive `*d*: [MS-DOS]	**Possible cause** The diskette in the specified drive is write-protected. **Action** Insert a diskette to which you can safely write and retry the operation. If you want to write on the write-protected diskette, remove the write-protect tab covering the notch in the diskette, and retry the operation.

Index

A

Active partition, 7-4
ANSI.SYS device driver, 6-10, A-9
ASCII files, 3-2
ASSIGN command, A-1
ATTRIB command, A-1
AUTOEXEC.BAT file, 2-3 6-3, 6-8, B-3, C-3
AUX, device name, 2-5

B

Backing up
  diskettes, 1-5
  hard disk, 5-5
  subdirectories, 5-6
BACKSPACE key, 1-4
BACKUP command, 5-5, A-1
BASIC program files, 3-3
Batch files
  creating, B-1
  displaying, B-3
  general, 3-2, B-1
  running, B-3
  terminating, B-4
Batch-processing commands, A-8
Baud rate, 6-5
Booting MS-DOS, 1-3
BREAK command, A-1
BREAK configuration command, 6-9, A-9
BUFFERS configuration command, 6-9, A-9

C

CACHE command, 7-5, 7-8, A-2
Canceling an operation, 2-4
CD command, 4-8, A-2
Changing directories, 4-8
Changing drives, 1-5
CHDIR command, 4-8, A-2
CHKDSK command, 5-6, A-2
CLS command, A-2
COM, file name extension, 3-3
COM port, 6-5
COM, device name, 2-5
Combining files, 5-2
COMMAND command, A-2
COMMAND.COM command processor, 1-8, 2-1, 6-9
CON, device name, 2-5
CONFIG.SYS, 3-3
CONFIG.SYS commands, A-9
CONFIG.SYS file, 6-6, C-3
Configuration commands, 6-9, A-9
Console, 2-5
COPY command, 1-9, 2-5, 5-1, 5-4, 7-9, A-2
COPY CON file creation, 6-7, B-2
Copy-protected software, 7-9
Copying,
  a disk, 1-5
  files, 5-1
  subdirectories, 5-3
  choosing appropriate command, 5-4
Correcting entries, 1-4
COUNTRY configuration command, 6-9, A-9, C-3
Creating directories, 4-4
CTRL ALT DEL keys, 1-3
CTRL PRTSC, 2-4, 4-8
CTTY command, A-2

D

Data bits, 6-5
Data files, 3-1
DATE command, A-2
DEBUG command, A-2
Default system configuration, 6-6
DEL command, 3-10, A-2
DEVICE configuration command, 6-9, 7-7, A-9
Device drivers, 6-9
Device names, 2-5
DIR command, 3-4, 3-8, 4-5, 4-8, 6-3, A-2
Directories
  changing, 4-8
  copying, 5-3
  current, 4-6, 4-9
  making, 4-4
  removing, 4-10
Directory
  list, 3-4
  root, 3-4
  markers, 4-10
  path, 4-6
  structure, 4-1
Disk caching, 7-5
Disk drive designators, 1-2, 6-9
DISKCOMP command, A-3

- DISKCOPY command, 1-5, 5-4, A-3
- Diskettes, formatting, 1-10
- DISKPACK command, 7-9, A-3
- Displaying file contents, 3-6
- Displaying output screen-by-screen, 6-4
- Drive designators, 1-2
- Drive, active, 1-4
- DRIVER.SYS device driver, 6-10, A-9
- DRIVPARM configuration command, 6-9, A-9

E

- ECHO batch-file command, A-8, B-2
- EDLIN program, A-3
- EEMS (Enhanced EMS), 7-6
- EMS (Expanded Memory Specification), 7-6
- End-of-file marker (CTRL Z), B-3
- EQUIP command, A-3
- ERASE command, 3-10
- Erasing files, 3-9
- Error messages, 2-2, D-1
- EXE, file name extension, 3-3
- EXE2BIN command, A-3
- Executable program files, 3-3
- Executing batch files, B-1
- EXIT command, A-3
- Expanded memory emulation, 6-6, 7-6
- Extended memory
  - CACHE, 7-6
  - compatibility, 7-8
  - VDISK.SYS, 6-11
  - VMEM.SYS, 7-6
- External commands, 2-1

F

- FC command, A-3
- FCBS configuration command, 6-9, A-9
- FDISK command, 1-7, 7-2, A-3
- File creation with COPY CON, 6-7, B-2
- File management, 3-1
- File name extensions
  - application programs, 3-3
  - BAS, 3-3
  - BAT, 3-3, B-1
  - changing, 3-8
  - COM, 3-3
  - EXE, 3-3
  - SYS, 3-3, 6-9
- File names
  - changing, 3-7
  - ficticious, 3-1
  - prohibited characters, 3-2
  - wild card, 3-5
  - with path names, 4-7
- File size, 3-4
- Files
  - copying, 5-1
  - deleting, 3-9
- FILES configuration command, 6-9, A-9
- Filtering output, 6-4
- FIND command, A-3
- FOR batch-file command, A-8
- FORMAT command, 1-8, 7-4, A-4
- Formatting
  - diskettes, 1-10
  - hard disks, 1-8
- Fragmented files, 7-9

G

- GOTO batch-file command, A-8
- GRAFTABL command, A-4
- GRAPHICS command, A-4
- GW-BASIC programming language, A-4

H

- Hard disk
  - backing up, 5-5
  - booting from, 1-9
  - formatting, 1-8
  - over 32MB, 7-1
  - partitioning, 1-7
  - preparation, 1-6
  - reorganizing, 7-9
- HDINIT command, A-4
- HDPARK command, A-4
- HELP command, 1-13, 2-2, 6-5, A-4
- Hidden files, 1-8, 2-1
- High-density drives, 1-2

I

- IF batch-file command, A-8
- Internal commands, 2-1

J

- JOIN command, 5-6, A-4

K

- Keyboards, international, C-1
- KEYBxx command, A-4, C-3

L

LABEL command, A-5
Large-disk support, 7-1
LASTDRIVE configuration command, 6-9, A-9
LCD panel, 7-12
LCDDEV device name, 2-5, 7-12, 7-14
LCDPANEL command, 7-12, A-5
LCDPANEL.SYS device driver, 2-5, 6-10, 7-12, 7-14, A-9
LIM specification, 7-6
LINK command, A-5
LOCK command, 7-15, A-5
Logical disk drives, 7-1
LPT device names, 2-5, 6-5

M

MD command, 4-4, A-5
Messages, D-1
MKDIR command, 4-4, A-5
MODE command, 2-5, 6-5, A-5
MORE command, 3-7, 6-4, A-5

N

National configurations, 1-8, C-1
Non-ASCII files, 3-2, 3-7
NUL, device name, 2-5

O

On-line help, 2-2
Output, redirecting, 6-3

P

Parallel ports (LPT), 2-5
Partitioning the hard disk, 1-7, 7-2
PATH command, 2-3, A-5
Path name, 4-6
PAUSE batch-file command, A-8
Performance graph, 7-13
Physical disk drives, 7-1
Ports, 2-5
PRINT command, A-5
Printer port, 2-5
Printers
  canceling port reassignments, 6-6
  reassigning ports, 6-5
  serial, 6-5
Printing screen contents, 2-4
PRN port, 6-5
PRN, device name, 2-5
Program files, 3-1, 3-3
PROMPT command, 4-8, 6-1, A-6

R

RAM disk, 6-10
RD command, 4-10, A-6
Rearranging output, 6-4
Reassigning ports, 6-5
Rebooting MS-DOS, 1-3
RECOVER command, A-6
Redirecting output, 6-3
REM batch-file command, A-8, B-2
Removing a directory, 4-10
REN command, 3-7
RENAME command, 3-7 A-6
Renaming files, 3-7
REPLACE command, A-
RESTORE command, 5- 5-7, A-6
Restoring backed up fil 5-7
RMDIR command, 4-10, A-6
Root directory, 3-4, 4-1, 4-9

S

Screen printing, 2-4
Scrolling, 2-4, 3-7
SELECT command, A-6, C-2
Serial ports (COM), 2-5
Serial printers, 6-3, 6-5
SET command, A-6
SHARE command, A-6
SHELL configuration command, 6-9, A-9
SHIFT batch-file command, A-8
SORT command, 6-4, A
SPARE command, A-7
SPEED command, 7-11, A-7
Stop bits, 6-5
Stopping an operation, 2-4
Subdirectories
  backing up, 5-6
  changing between, 4-8
  copying, 5-3
  general, 4-1, 4-3, 6-1
SUBST command, 5-6, A-7

Supplemental programs diskette, 1-3, 1-9
SYS command, A-7
SYS, file name extension, 3-3, 6-9
System files, 1-8, 2-1, 3-3
System messages, 2-2, D-1

T

Target disk, 1-5
Target file, 5-2
Text files, 3-2
TIME command, A-7
Time prompt, 1-4
TREE command, 4-7, A-7
TYPE command, 3-6, A-7, B-3

U

Unlocking the system, 7-17
User files, MS-DOS, 1-3, 1-9

V

VDISK.SYS device driver, 6-10, 7-8, A-9
VER command, A-7
VERIFY command, A-7
Virtual disk, 6-6, 6-10
VMEM command, 7-8, A-8
VMEM.SYS device driver, 6-10, 7-6, A-9
VOL command, 1-8, A-8
Volume label, 1-8, 1-11, 3-4

W

Wild cards
  asterisk, 1-9, 3-6
  combinations, 3-6
  combining files, 5-3
  in file names, 3-5
  question mark, 3-5
  with DEL command, 3-10
  with REN command, 3-8

X

XCOPY command, 5-3, A-8